Felix Brandt
Piotr Faliszewski (Eds.)

Workshop Notes

The Fourth International Workshop on Computational Social Choice

COMSOC 2012

September 11—13, 2012

AGH University of Science and Technology
Kraków, Poland

Artificial Intelligence

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

CLICO 20 lat

Preface

Computational social choice is a relatively new research field whose goal is, on one hand, to provide computational and algorithmic analysis of social choice and group decision making settings, and, on the other, to apply methods and tools of social choice in the context of computer science. Thus, naturally, computational social choice is a very multidisciplinary field, spanning computer science, economics, and political science, with an active exchange of ideas between all these fields. A classic example of such an exchange is the concept of a voting-based Internet meta-search engine: Search engines (the voters) rank websites (the candidates) with respect to a given query, the rankings are aggregated using some voting rule, the final result is presented to the user. The difficulty in implementing this intuitive, beautiful idea lies in the fact that we have to pick a voting rule that will lead to desirable aggregated results and, at the same time, can be efficiently computed. Moreover, it should be difficult for users to skew the results of these "online elections" (for example, to promote or bury some website). Thus, to implement the idea, one has to have a good understanding of the axiomatic properties of voting rules (and group decision-making in general), a good understanding of algorithms for computing voting rules (or their approximate variants), as well as a good understanding of methods of manipulation and the practical and theoretical hardness of implementing manipulative attacks. Research in computational social choice addresses these—and many other—issues using a variety of methods ranging from computational complexity theory, through computational and empirical experiments, to pure theoretical social choice.

This volume contains the workshop notes of COMSOC-2012, the Fourth International Workshop on Computational Social Choice, hosted on 11–13th September 2012 by the Department of Computer Science at the AGH University of Science and Technology in Kraków, Poland. The COMSOC workshop series started in 2006 in Amsterdam and since then continued biennially, with the second workshop held in Liverpool (2008) and the third one held in Düsseldorf (2010). The goal of the workshop series is to bring together people from various communities that contribute to computational social choice (computer scientists, economists, and political scientists) and to provide a forum for them to exchange ideas, discuss their research, and discover new approaches.

We received 54 submissions of which we accepted 38. Virtually all of the submissions were of very high quality and highly relevant to the theme of the workshop, making the selection process particularly difficult. As in the previous editions, both new papers and papers already presented or accepted at some other venue (for example, at conferences with formal proceedings such as AAAI, IJCAI, or AAMAS) were accepted. Each paper was reviewed by three program committee members and/or outside referees. For each paper included in this volume, its copyright stays with the authors.

As opposed to previous years, this time we have decided not to organize a separate tutorial day, but rather to focus on a single topic with an extended tutorial at the beginning of the workshop's first day. Our chosen topic is parametrized complexity theory, a theory that is employed ever more frequently in computational social choice papers:

- Prof. Rolf Niedermeier, Technische Universität Berlin
  Parameterized Complexity Analysis for Social Choice Problems

We will also have four invited talks, delivered by some of the most prominent researchers working on (computational) social choice:

- Prof. Michel Le Breton, Toulouse School of Economics
  Simple Games and the Probability of Casting a Decisive Vote.

- Prof. Clemens Puppe, Karlsruhe Institute of Technology
  Majority Voting over Interconnected Propositions: The Condorcet Set.
- Prof. Craig A. Tovey, Georgia Institute of Technology
  Computational Methods for the Spatial Model of Social Choice.
- Prof. Gerhard Woeginger, Eindhoven University of Technology
  Coalitions in Hedonic Games.

We are very grateful to all the people that helped in preparing the workshop. Foremost, we would like to thank Ulle Endriss and Jérôme Lang, who have organized the first COMSOC workshop and, since then, have put tremendous effort into maintaining the community, keeping the workshop series going, organizing various other computational social choice events, obtaining funding, and promoting the field in the wider scientific community. We also are very grateful to all the program committee members, reviewers, local organizing committee members, the sponsors, and—last but not least—to our invited guests and paper authors, who all contribute to making COMSOC 2012 a successful event.

F.B. & P.F.
Munich & Kraków, August 2012

Program Committee

Craig Boutilier	University of Toronto
Felix Brandt	Technische Universitaet Muenchen
Edith Elkind	Nanyang Technological University
Ulle Endriss	University of Amsterdam
Piotr Faliszewski	AGH University of Science and Technology
Bettina Klaus	University of Lausanne
Jerome Lang	Université Paris-Dauphine
Christian List	London School of Economics
Hannu Nurmi	University of Turku
Maria Silvia Pini	University of Padova
Ariel Procaccia	Carnegie Mellon University
Jeffrey S. Rosenschein	The Hebrew University of Jerusalem
Joerg Rothe	Universitaet Duesseldorf
Tuomas Sandholm	Carnegie Mellon University
Michael Trick	Carnegie Mellon University
Virginia Vassilewska-Williams	University of California, Berkeley
Toby Walsh	NICTA and UNSW
Michael Wooldridge	University of Liverpool
Lirong Xia	Harvard University
Makoto Yokoo	Kyushu University
William S. Zwicker	Union College Mathematics Department

Additional Reviewers

Aziz, Haris

Ball, Michael
Ballester, Miguel
Baumeister, Dorothea
Brill, Markus
Budinich, Michele

Dorn, Britta

Erdelyi, Gabor

Gehrlein, William

Harrenstein, Paul
Huang, Chien-Chung

Iwasaki, Atsushi

Klamler, Christian

Lev, Omer
Lu, Tyler

Maudet, Nicolas
Meir, Reshef
Moulin, Herve

Nguyen, Nhan-Tam

Polukarov, Maria
Pyrga, Evangelia

Rey, Anja
Roos, Magnus
Rossi, Francesca
Roth, Aaron

Sakurai, Yuko
Salehi-Abari, Amirali
Salvagnin, Domenico
Schend, Lena
Schlotter, Ildikó
Seedig, Hans Georg
Skowron, Piotr
Sui, Xin

Tang, Pingzhong
Todo, Taiki

Ueda, Suguru

Venable, K. Brent

Zuckerman, Michael

Sponsors

COMSOC 2012 is generously sponsored by:

Artificial Intelligence Journal

Institute for Logic, Language and Computation, University of Amsterdam

CLICO Sp. z o.o.

The workshop is supported by the Department of Computer Science of the AGH Univesity of Science and Technology in Kraków, Poland.

Table of Contents

Computing Socially-Efficient Cake Divisions

Yonatan Aumann, Yair Dombb and Avinatan Hassidim

Abstract

We consider a setting in which a single divisible good ("cake") needs to be divided between n players, each with a possibly different valuation function over pieces of the cake. For this setting, we address the problem of finding divisions that maximize the *social welfare*, focusing on divisions where each player needs to get one contiguous piece of the cake. We show that for both the utilitarian and the egalitarian social welfare functions it is NP-hard to find the optimal division. For the utilitarian welfare, we provide a constant factor approximation algorithm, and prove that no FPTAS is possible unless P=NP. For egalitarian welfare, we prove that it is NP-hard to approximate the optimum to any factor smaller than 2. For the case where the number of players is small, we provide an FPT (fixed parameter tractable) FPTAS for both the utilitarian and the egalitarian welfare objectives.

1 Introduction

Consider a town with a central conference hall, erected by the municipality for the benefit of the townspeople. Different people and organizations wish to use the hall for their events, each for a possibly different duration. Furthermore, each such event may have its preferences and constraints on the times when it can take place, e.g. only in the evenings, on weekends, prior to some date, etc. How should the municipality allocate the hall to the different events? How do we compute the allocation that maximizes the social welfare provided by this common resource?

A natural setting for analyzing the above problem is that of *cake cutting*, where a single divisible good needs to be divided between several players with possibly different preferences regarding the different parts of the good, or "cake". The cake cutting problem was first introduced in the 1940's by Steinhaus [Ste49], where the goal was to give each of the n players "their due part", i.e. a piece worth at least $\frac{1}{n}$ of the entire cake by their own measure. (In the cake cutting literature, this fairness requirement is termed *proportionality*.) Since then, other objectives have also been considered, with the majority of them requiring that the division be "fair", under some definition of fairness (e.g. envy-freeness).

Here, we address the fundamental problem of *maximizing social welfare* in cake cutting. Given a shared resource, the valuation functions of the players for this resource, and a social welfare function, the problem is to find an allocation that maximizes the welfare. Maximizing social welfare has been previously considered for dividing a set of discrete indivisible items, each of which must be given in whole to one player. Here, we consider the problem with a *single, continuously divisible good*, and furthermore focus on the case where each player needs to get a *single contiguous piece* of the good. The contiguity requirement is natural in many settings, e.g. dividing time (as in the example above), spectrum, and real-estate.

Results. We show that the problems of maximizing utilitarian and egalitarian welfare are both NP-hard in the strong sense. For egalitarian welfare, we further show that it is hard to approximate the optimum to any factor smaller than 2.

For utilitarian welfare, we provide a constant-factor approximation algorithm (note that the strong NP-hardness result implies that no FPTAS exists for the problem). Specifically, our algorithm finds a division with utilitarian welfare within $8 + o(1)$ of the optimum, in polynomial time. We also show that approximating both the utilitarian and egalitarian welfare is fixed-parameter-tractable with regards to the parameter n (the number of players).

Finally, we consider the case where the contiguity requirement is dropped, i.e. each player may get a *collection* of intervals. For this setting, we show that the situation varies greatly depending on the model of input. When the valuations are given explicitly to the algorithm, and are piecewise constant, the problem can be solved in polynomial time. However, if the algorithm has only oracle access to the valuations, then it is impossible to do any better than an n-factor approximation, even if the valuations themselves are piecewise uniform.

Due to space constraints, many of the proofs are deferred to the full version of the paper.

Related Work. The problem of maximizing egalitarian welfare when allocating a set of indivisible goods has been extensively considered in the last 15 years [Woe97, AAWY98, BS06, CCK09]. The currently known best algorithms are a polynomial-time algorithm achieving an approximation factor of $O(\sqrt{n}\log^3 n)$ [AS07], and an algorithm obtaining $\tilde{O}(n^\epsilon)$ approximation in time $n^{O(1/\epsilon)}$, for any $\epsilon = \Omega(\frac{\log\log n}{\log n})$ [CCK09]. Hardness of approximation for this problem, however, is proven only for a factor of 2 or less [BD05]. Better approximation guarantees are known for more restricted settings, e.g. when valuations are restricted to having only one possible non-zero value for each item [BS06, Fei08]. Envy minimization in this setting has also been considered in [LMMS04], which showed hardness results as well as an FPTAS for the case of players with identical preferences. Unlike this body of work, which considers a *non-ordered set* of indivisible items, here we consider a single divisible item, and furthermore require that each player obtain a single contiguous piece of this good.

Cake cutting problems were first introduced in the 1940's [Ste49], and were studied in many variants since then. Various algorithms were proposed for the problem, including a number of "moving knife" algorithms, which perform an infinite number of valuations by continuously moving a knife over the cake (for some examples, see [Str80, EP84] and [BT95]). In addition to the algorithmic results, there has also been work on existence theorems [DS61, Str80], lower bounds for the complexity of such algorithms ([SW03, Str08, Pro09], to mention just a few), and a number of books on the subject, e.g. [BT96, RW98].

The issue of social welfare in cake cutting was first considered in Caragiannis et al. [CKKK09] which aimed to quantify the degradation in social welfare that may be caused by different fairness requirements; the same question was studied for connected pieces in [AD10]. Guo and Conitzer [GC10], and Han et al. [HSTZ11] study the utilitarian welfare achievable by truthful mechanisms for dividing a set of divisible goods, a setting very similar to a cake with piecewise-constant valuations and non-connected pieces. Cohler et al. [CLPP11] study utilitarian welfare maximization under the envy-freeness requirement (with non-connected pieces). Bei et al. [BCH+12] consider a similar question, but with connected pieces, and with proportionality replacing envy-freeness. Also related is the work of Zivan [Ziv11] which suggests a way for increasing utilitarian welfare using trust.

2 Model and Definitions

Valuation Functions. In our model, the cake is represented by the interval $[0, 1]$. Each player $i \in [n]$ (where $[n] = \{1, \ldots, n\}$) has a non-atomic (additive) measure $v_i(\cdot)$, mapping each measurable subset of $[0, 1]$ to its value according to player i. For most of this work, we are only interested in a value of *intervals* in $[0, 1]$, and thus simply write $v_i(a, b)$ for the value of the interval between a and b. (Note that since v_i is non-atomic, single points have zero value, and we need not worry about the boundary points a and b themselves.)

We also assume, as common in the cake-cutting literature, that the valuations are *normalized*, i.e. that $v_i(0, 1) = 1$ for every player i. However, our results hold (with small modifications to the algorithms or complexity) for arbitrary valuations as well.

Social Welfare Functions. We consider two prominent social welfare functions, whose aim is to measure how good each division is for the *whole society*. Let x be a division (to be formally defined shortly); we write $u_i(x)$ to express the value player i obtains from the piece she receives in x. The *utilitarian welfare* is defined as the sum of utilities, and we denote $u(x) = \sum_{i \in [n]} u_i(x)$. The *egalitarian welfare* is defined as the utility of the worst-off player, and we denote $eg(x) = \min_{i \in [n]} u_i(x)$.

Connected Divisions. In this work, we focus on divisions in which every player gets a (disjoint) *single interval* of the cake. Formally, a connected division of the cake $[0, 1]$ between n players can be defined as a vector $x = (x_1, \ldots, x_{n-1}, \pi) \in [0, 1]^{n-1} \times S_n$ (where S_n is the set of all the permutations of $[n]$), having $x_1 \leq x_2 \leq \cdots \leq x_{n-1}$. This is interpreted as making $n - 1$ cuts in positions $x_1, \ldots, x_{n-1}$, and allocating the n resulting intervals to the players in the order determined by the permutation π. Note that the space X of all such divisions is compact; in addition, both utilitarian and egalitarian welfare functions are continuous in X (as the players' valuation functions are all non-atomic). Therefore, for each of these welfare functions there exists a division that maximizes the welfare.

Our main problem is thus the following: *given the players' valuations, what is the (connected) division that maximizes welfare?* Since the two welfare functions considered here obtain maxima in the divisions space, the problem is indeed well-defined. For the analysis of these problems, it is useful to consider their decision versions, defined as follows.

Connected Utilitarian Optimum (CUO)	
Instance:	A set $\{v_i\}_{i=1}^n$ of non-atomic measures on $[0, 1]$, and a bound B.
Problem:	Does there exist a connected division x having $u(x) \geq B$?

Connected Egalitarian Optimum (CEO)	
Instance:	A set $\{v_i\}_{i=1}^n$ of non-atomic measures on $[0, 1]$, and a bound B.
Problem:	Does there exist a connected division x having $eg(x) \geq B$?

Complexity and Input Models. In order to analyze the complexity of our problems, we must first define how the input is represented. In most of the cake cutting literature, the mechanism is not explicitly given the players' valuation functions; instead, it can *query* the players on their valuations (see e.g. [EP84, RW98, Str08]). Typically, two types of queries are allowed. In the first, a player i is given points $0 \leq a \leq b \leq 1$ and is required to return the value $v_i(a, b)$. In the second type of query, a player i is given a point $a \in [0, 1]$ and a value x and is required to return a point b such that $v_i(a, b) = x$; we denote this by $v_i^{-1}(a, x)$.[1]

In contrast, some more recent works (e.g. [CLPP10, CLPP11, BCH$^+$12]) consider a model in which the players give complete descriptions of their valuations to the mechanism. In this case, it is usually assumed that the functions have some simple structure, so they can be represented succinctly. Specifically, for each player i, let $\nu_i : [0, 1] \to [0, \infty)$ be a *value density function*, such that

$$v_i(X) = \int_X \nu_i(x) dx$$

for every measurable subset $X \in [0, 1]$. Following [CLPP10], we say that a valuation function $v_i(\cdot)$ is *piecewise-constant* if its value density function $\nu_i(\cdot)$ is a step function, i.e. if $[0, 1]$

[1]Note that using only one type of query it is possible to give approximate answers (in polynomial time) to queries of the other type using binary search.

can be partitioned into a *finite* number of intervals such that ν_i is constant on each interval. If, in addition, there is some constant c_i such that $\nu_i(\cdot)$ can only attain the values 0 or c_i, we say that $v_i(\cdot)$ is *piecewise-uniform.*[2] Any piecewise-constant valuation function $v_i(\cdot)$ can be therefore represented by a finite set of subintervals of $[0, 1]$ together with the value ν_i attains in each interval.

Our hardness results show that both of the decision problems above are computationally hard, even when the valuation functions are of the simplest type—piecewise-uniform—and are given explicitly to the mechanism. In contrast, our positive algorithmic results hold also for the more general oracle model. The complexity of our algorithms in this case depends on the number of players n and additionally on a precision parameter ϵ.

The Discrete Variants. A convenient preprocessing step in our algorithms will be reducing our problems into ones that are purely combinatorial. More precisely, we consider discrete analogues of the problems, where one is additionally given a set of points in $[0, 1]$, and is only allowed to make cuts at points from this set (and not anywhere in $[0, 1]$). An alternative interpretation is to consider, instead of a continuous cake, a *sequence of indivisible items*; a connected division in this setting gives each player a *consecutive subsequence* of these items. The discrete variants of our problems are defined as follows:

Discrete Connected Utilitarian Optimum (Discrete-CUO)	
Instance:	A sequence $A = (a_1, \ldots, a_m)$ of items, a set $\{v_i\}_{i=1}^n$ of valuation functions of the form $v_i : A \to \mathbb{R}^+$, and a bound B.
Problem:	Does there exist a connected division x having $u(x) \geq B$?

Discrete Connected Egalitarian Optimum (Discrete-CEO)	
Instance:	A sequence $A = (a_1, \ldots, a_m)$ of items, a set $\{v_i\}_{i=1}^n$ of valuation functions of the form $v_i : A \to \mathbb{R}^+$, and a bound B.
Problem:	Does there exist a connected division x having $eg(x) \geq B$?

Our hardness results apply to these "cleaner" problems as well. We note that if we drop the contiguity requirement, allowing players to get any disjoint subsets of A, maximizing utilitarian welfare becomes trivial (give each item to the player who values it the most). In contrast, maximizing egalitarian welfare (in the discrete setting with non-connected pieces) is known to be a hard problem [BD05] and has been studied extensively (e.g. [AS07, CCK09]).

3 Approximation Algorithms

In this section we present algorithms that return a division that is guaranteed not to be too far from the social optimum. Throughout this section we assume that the algorithms operate in the (more-general) oracle model. We note that if the valuation functions are given explicitly, and are simple enough (in particular, if they are piecewise-constant), the answer to each oracle query can be computed in time polynomial in the input size.

3.1 The Discretization Procedure

As we have previously mentioned, it is often useful to reduce the continuous cake into a sequence of discrete items. We now show that this can indeed be done in a time-efficient manner, and without too much harm to the maximum obtainable welfare.

[2] Note that in this case the constant c_i is uniquely determined by the total fraction of $[0, 1]$ in which $\nu_i(x) \neq 0$, since we require that the valuation of the entire cake should be 1.

The *Discretization Procedure* receives a cake instance and a parameter ϵ, and produces a set of cut positions that partition the cake into a set of items. We start with the set $C = \{0\}$ of cut points. At each step, let a be the position of the last (rightmost) cut in C. The procedure asks each player i for the leftmost point b_i such that the $v_i(a, b_i) = \epsilon$; it then adds the leftmost of these points to C, and repeats the process. When $v_i(a, 1) \leq \epsilon$ for all players i, the procedure adds the point 1 to C, and halts.

Note that the set of cuts C produced by the algorithm induces a sequence of items. Specifically, let $0 = c_0 < c_1 < \cdots < c_m = 1$ be the cut points in C; then, for each $1 \leq j \leq m$ create an item a_j with value $v_i(a_j) = v_i(c_{j-1}, c_j)$ for player $i \in [n]$.

The following lemma, whose proof we omit due to space constraints, establishes that the set C can be computed efficiently, and that we do not lose much utilitarian welfare by restricting our cuts positions to C. A similar claim also holds for egalitarian welfare.

Lemma 1. *Let $\{v_i(\cdot)\}_{i\in[n]}$ be a cake instance with n players, and consider some precision parameter ϵ. Then:*

1. *The discretization procedure terminates on this instance in time $O(n^2/\epsilon)$.*
2. *Let x be a division of the original cake; then there exists a division y making cuts only at points in the set C returned by the procedure, and having $u(y) \geq u(x) - (n-1)\epsilon$.*

3.2 Approximating the Utilitarian Welfare

We now present an approximation algorithm for the problem of maximizing utilitarian welfare; the approximation ratio achieved by our algorithm is $8\big(1 + (n-1)\epsilon\big)$, where ϵ is a precision parameter, and the running time of the algorithm is polynomial in n and in $1/\epsilon$. As a first step, the algorithm uses the Discretization Procedure to obtains a set A of m discrete items. We now describe how to approximate the optimal utilitarian welfare for this new instance. The algorithm returns a set $\{(s_i, t_i)\}_{i\in[n]}$, where s_i is the beginning index of i's piece, and t_i is its end index. We also use the notation (s, t) to refer to the consecutive sequence of items $\{s, s+1, \ldots, t-1, t\}$; hence, e.g. $v_i(s, t) = \sum_{j=s}^{t} v_i(j)$.

Algorithm 1: Discrete Utilitarian Welfare Approximation

Data: For each player $i \in [n]$ a vector of valuations $v_i : [m] \to \mathbb{R}^+$.
begin
  $\forall i \in [n] : s_i \longleftarrow 0\ ,\ t_i \longleftarrow 0$
  for $t = 1, \ldots, m$ **do**
    while $\max_{k\in[n], s\leq t} \Big\{ v_k(s,t) - 2\Big(v_k(s_k, t_k) + V_{-k}(s,t)\Big)\Big\} \geq 0$
    do
      $k', s' \longleftarrow$ arguments maximizing the expression
      $s_{k'} \longleftarrow s'\ ,\ t_{k'} \longleftarrow t$
      $(s_i, t_i) \longleftarrow (0, 0)$ for all i with $s_i \geq s'$
      $t_i \longleftarrow s' - 1$ for i with $s_i < s' \leq t_i$
  return $\big\{(s_i, t_i)\big\}_{i\in[n]}$

Our algorithm for the discretized instance works iteratively, where in the t-th iteration it finds a good division for the first t items. We begin with the trivial null allocation of 0 items. Assuming that we have a good allocation for the first $t-1$ items, and for all $s \leq t$ and $k \in [n]$, we consider the *cost* of giving items s through t to player k. This cost is comprised of two components. The first component is the value of a piece (s_k, t_k) that player k may currently own, and has to give up in order to get the new piece (s, t). The second component is the sum of values that the other players to which the items s through t are assigned obtain from these items. We denote this second component by $V_{-k}(s, t)$. We only give the segment (s, t) to player k if her total value $v_k(s, t)$ for this segment is at least twice the cost of giving her this segment. We continue trying to find a player k' and a segment (s', t) ending at item t whose value exceeds twice the cost, and make changes until there are no such player and segment, at which point we move on to the next item $t + 1$.

Observe that in the algorithm, each interval (s, t) can be given to player i at most once; this immediately implies that the running time of the algorithm is polynomial in the number of players n and number of items m. For analyzing the approximation ratio of the algorithm, we use indicator variables x_i^j, for $i \in [n]$ and $j \in [m]$. At each step in the algorithm, we will have $x_i^j = 1$ if and only if player i owned the item j at some point until now.

Lemma 2. *At any iteration t of the above algorithm, we have*

$$\sum_{i\in[n]} v_i(s_i, t_i) \leq \sum_{i\in[n]} \sum_{j\in[m]} x_i^j \cdot v_i(j) \leq 2 \cdot \sum_{i\in[n]} v_i(s_i, t_i)$$

(where the values are as in the end of the t-th iteration).

Proof. The first inequality trivially holds, and we prove the second by induction on t. The second inequality clearly holds at the beginning of the step $t = 1$; we show that if it holds at the beginning of some step t, then it must still hold at the end of this step.

At the beginning of the t-th step, item t is unallocated. If the while loop was not executed even once in this iteration, none of the expressions $\sum_{i\in[n]} \sum_{j\in[m]} x_j^i \cdot v_i(j)$ and $\sum_{i\in[n]} v_i(s_i, t_i)$ have changed, and the claim still holds. Otherwise, consider some iteration of the while loop. In such an iteration, the increase in $\sum_{i\in[n]} \sum_{j\in[m]} x_i^j \cdot v_i(j)$ is upper-bounded by $v_{k'}(s', t)$. The expression $\sum_{i\in[n]} v_i(s_i, t_i)$ also gains $v_{k'}(s', t)$, but in addition loses $v_k(s_k, t_k) + V_{-k}(s, t)$; however, the while loop condition ensures that $v_{k'}(s', t) - \big(v_k(s_k, t_k) + V_{-k}(s, t)\big) \geq \frac{1}{2} \cdot v_{k'}(s', t)$. Therefore, the increase to the right-hand side of the inequality is at least as large as that of the left-hand side, and the inequality is maintained. Since this holds for every iteration of the while loop, this still holds at the end of step t, as required. □

Theorem 1. *Algorithm 2 returns an 8-approximation of the discrete utilitarian optimum.*

Proof. Fix a discrete cake instance. Let $\big\{(s_i^A, t_i^A)\big\}_{i\in[n]}$ be the final output of Algorithm 1 on this instance, and let $\big\{(s_i^*, t_i^*)\big\}_{i\in[n]}$ be the optimal division for this instance. Denote by $OPT = \sum_{i\in[n]} v_i(s_i^*, t_i^*)$ the utilitarian welfare achieved by the optimal division.

For every player k, consider the iteration t_k^*, in which the rightmost item given to k in the optimal division was first considered. Let (s_k', t_k') be the segment given to player k at the end of this iteration. When iteration t_k^* ends, it has to be that

$$v_k(s_k^*, t_k^*) \leq 2\big(v_k(s_k', t_k') + V_{-k}(s_k^*, t_k^*)\big)$$

(where $V_{-k}(s_k^*, t_k^*)$ is with respect to the division set by the algorithm at this point). Note that $v_k(s_k', t_k') = \sum_{j=s_k'}^{t_k'} x_j^k \cdot v_k(j)$ and that $V_{-k}(s_k^*, t_k^*) \leq \sum_{j=s_k^*}^{t_k^*} \sum_{i\neq k} x_j^i \cdot v_i(j)$. Combining all this, we get

$$
\begin{aligned}
OPT = \sum_{k\in[n]} v_k(s_k^*, t_k^*) &\le \sum_{k\in[n]} 2\cdot\Big(\sum_{j=s'_k}^{t'_k} x_j^k\cdot v_k(j) + \sum_{j=s_k^*}^{t_k^*}\sum_{i\neq k} x_j^i\cdot v_i(j)\Big) \\
&= 2\cdot\Big(\sum_{k\in[n]}\sum_{j=s'_k}^{t'_k} x_j^k\cdot v_k(j) + \sum_{k\in[n]}\sum_{j=s_k^*}^{t_k^*}\sum_{i\neq k} x_j^i\cdot v_i(j)\Big) \\
&\le 2\cdot\Big(\sum_{k\in[n]}\sum_{j\in[m]} x_j^k\cdot v_k(j) + \sum_{k\in[n]}\sum_{j\in[m]} x_j^k\cdot v_k(j)\Big) \\
&= 4\cdot\sum_{k\in[n]}\sum_{j\in[m]} x_j^k\cdot v_k(j) \le 8\cdot\sum_{i\in[n]} v_i(s_i^A, t_i^A)\ .
\end{aligned}
$$

The second inequality holds since for every $k \neq k'$ the segments (s_k^*, t_k^*) and $(s_{k'}^*, t_{k'}^*)$ are disjoint, as $\{(s_i^*, t_i^*)\}_{i\in[n]}$ is a division. The last inequality follows from Lemma 2. □

Combining the guarantees for the Discretization Procedure and for Algorithm 1 we get:

Corollary 2. *For every $\epsilon > 0$, it is possible to find a division achieving utilitarian welfare within $8\big(1+(n-1)\epsilon\big)$ of the optimum in time polynomial in n and $1/\epsilon$.*

3.3 Fixed-Parameter Tractable Approximations

Suppose that we have a relatively small number of players n, but that the social efficiency of the division is of much importance. We show that divisions that are within a factor of $1+\epsilon$ of the social optimum (for both utilitarian and egalitarian welfare) can be computed in time exponential in the number of players, but polynomial in $\frac{1}{\epsilon}$.[3] Using the terminology of the theory of Parametrized Complexity [DF99] we say that these approximations are *fixed-parameter tractable*. Both of these algorithms are based on dynamic programming; the full proofs can be found in the full version of the paper.

Theorem 3. *For every $\epsilon > 0$, it is possible to find a division achieving utilitarian welfare within $1+\epsilon$ of the optimum in time $2^n \cdot poly(n, \frac{1}{\epsilon})$.*

Theorem 4. *For every $\epsilon > 0$, it is possible to find a division achieving egalitarian welfare within $1+\epsilon$ of the optimum in time $2^n \cdot n \cdot \log_2\left(\frac{n}{\epsilon}\right)$.*

4 Hardness

We show that all of the four problems defined in Section 2 are NP-complete in the strong sense. Note that membership in NP is straightforward, as a division achieving the required welfare can serve as a witness for that instance; we thus concentrate on proving hardness.

We prove that CEO is strongly NP-complete and hard to approximate to a factor of $2-\epsilon$ for any $\epsilon > 0$, using a reduction from the classic problem of 3DM [GJ79]. In this problem, one is given three sets X, Y, Z of cardinality n each, as well as a set $E \subseteq X \times Y \times Z$, and needs to determine if there exists a subset $E' \subseteq E$ of cardinality n that covers X, Y and Z.

Our reduction borrows its main ideas from the proof of Bezáková and Dani [BD05] for non-connected divisions in the discrete setting, which itself uses ideas from Lenstra, Shmoys and Tardos [LST90]. However, the adjustment to the continuous setting with connected divisions is somewhat intricate and needs to be done carefully.

[3] Recall that we assume the oracle model; if the valuation functions are given explicitly, we also have polynomial dependence on the size of the input.

Theorem 5. CEO *and* DISCRETE-CEO *are NP-complete in the strong sense. Furthermore, for every* $\epsilon > 0$ *there is no* $(2-\epsilon)$ *approximation for either of the problems, unless P=NP.*

This holds even if the valuation functions of the players are piecewise-uniform, and are given explicitly to the algorithm.

Proof. We show a polynomial-time reduction from 3DM to CEO. Let X, Y, Z and $E \subseteq X \times Y \times Z$ be an input to 3DM. We construct a set of piecewise-constant valuations and a bound B as an input for CEO; this instance can be transformed into an equivalent one with piecewise-uniform valuations.

For convenience, we take the cake to be the interval $\big[0, 2|E|\big]$ rather than $[0,1]$. We will think of the cake as being sectioned into $|E|$ "sections" of length 2, where the right half of each section is used for separation from the next section.[4] The set of players we create has players of three types: "triplet players", "ground sets players" and "separation players". In what follows we describe the valuation functions of all the players, by their type; for the bound, we set $B = \frac{1}{|E|}$.

- **Triplet Players:** We create a player for every $z \in Z$. For each $e_i \in E$ such that z appears in the triplet e_i, the player created for z has value of $\frac{1}{2|E|}$ for each of the intervals $\big(2(i-1), 2(i-1)+\frac{1}{4}\big)$ and $\big(2(i-1)+\frac{3}{4}, 2(i-1)+1\big)$ in the left half of the i-th section.

  Denote by m_z the number of such triplets e_i in E. To keep the value of the entire cake at 1 for each player, we will divide the missing value $1 - \frac{m_z}{|E|}$ between the right halves of all sections. Specifically, player z will additionally have value $\frac{|E|-m_z}{2|E|^2}$ for every interval $\big(2(j-1)+\frac{6}{5}, 2(j-1)+\frac{7}{5}\big)$ and $\big(2(j-1)+\frac{8}{5}, 2(j-1)+\frac{9}{5}\big)$, for all $1 \le j \le |E|$.

- **Ground Sets Players:** For $x \in X$, let m_x be the number of triplets in E in which x appears. We create $m_x - 1$ identical players for x. For every $e_i \in |E|$ such that x appears in e_i, all of x's players will have valuation of $\frac{1}{|E|}$ for the interval $\big(2(i-1)+\frac{1}{4}, 2(i-1)+\frac{1}{2}\big)$ in the left half of the i-th section. Again, in order to complement these valuations to 1, they will also assign a value of $\frac{|E|-m_x}{2|E|^2}$ for each of the intervals $\big(2(j-1)+\frac{6}{5}, 2(j-1)+\frac{7}{5}\big)$ and $\big(2(j-1)+\frac{8}{5}, 2(j-1)+\frac{9}{5}\big)$, for all $1 \le j \le |E|$.

  We similarly create $m_y - 1$ identical players for every $y \in Y$. For each $e_i \in E$ in which y appears we have these players give value of $\frac{1}{|E|}$ to the interval $\big(2(i-1)+\frac{1}{2}, 2(i-1)+\frac{3}{4}\big)$, and complement this by giving value of $\frac{|E|-m_y}{2|E|^2}$ to each of the intervals $\big(2(j-1)+\frac{6}{5}, 2(j-1)+\frac{7}{5}\big)$ and $\big(2(j-1)+\frac{8}{5}, 2(j-1)+\frac{9}{5}\big)$, for all $1 \le j \le |E|$.

- **Separation Players:** We finally create $3|E|$ separation players. For every segment $1 \le i \le |E|$ we have a player s_{3i-2} have valuation of 1 for the interval $\big(2(j-1)+1, 3(j-1)+\frac{6}{5}\big)$, another player s_{3i-1} have valuation 1 for $\big(2(i-1)+\frac{7}{5}, 2(i-1)+\frac{8}{5}\big)$, and a third player s_{3i} have valuation 1 for $\big(2(i-1)+\frac{9}{5}, 2(j-1)+2\big)$.

Figure 1 illustrates the structure of the preferences in one segment. In this example, we consider some triplet $e_i = (x_j, y_k, z_\ell) \in E$, and show the section of the cake created for it, with the preferences of the players who desire some piece of it.

It is straightforward to observe that the construction above can be carried out in polynomial time. Also, all the numbers created in this instance can be represented by a number of bits logarithmic in the input size.

[4]Indeed, the last section needs not have this "separation half"; however, we leave it there in order to treat it identically to all the other sections.

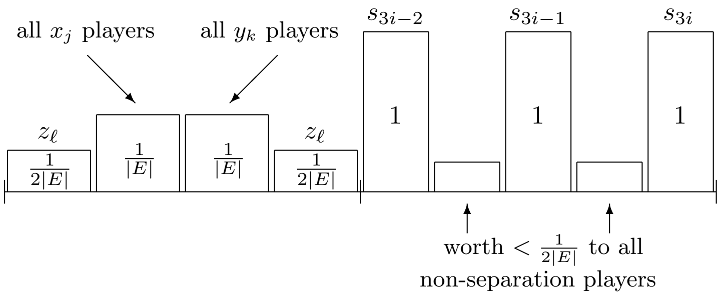


Figure 1: The valuations of the players for the section created for $e_i = (x_j, y_k, z_\ell) \in E$. Note that there are $m_{x_j} - 1$ identical players for x_j and $m_{y_k} - 1$ identical players for y_k.

Due to space constraints, we defer the correctness proof for this construction, as well as the adjustment for piecewise-uniform valuations, to the full version of the paper.

The proof for DISCRETE-CEO is analogous, and can easily be obtained by a straightforward partitioning of the cake created in the reduction into discrete indivisible chunks. □

We use a reduction from DISCRETE-CEO to prove the hardness of maximizing utilitarian welfare. The proof is again deferred to the full version due to space constraints.

Theorem 6. CUO *and* DISCRETE-CUO *are NP-complete in the strong sense.*

This holds even if the valuation functions of the players are piecewise-uniform, and are given explicitly to the algorithm.

The strong NP-hardness of CUO and DISCRETE-CUO implies the following corollary:

Corollary 7. *There is no FPTAS for either* CUO *nor* DISCRETE-CUO*.*

5 Welfare Maximization with Non-Connected pieces

In this section we analyze the problem of welfare maximization when each player may get a *collection* of intervals. We first show that if the valuation functions are piecewise-constant and are given explicitly to the algorithm, the problem can be easily solved in polynomial time using a linear program almost identical to the one used by Cohler et al. [CLPP11]; the details of the proof are can be found in the full version of the paper.

Theorem 8. *Given a set of n piecewise-constant valuation functions (i.e. for each $i \in [n]$ the list of intervals in which the value density function attains different values, along with the value for each such interval), it is possible to find a division maximizing the utilitarian (resp. egalitarian) welfare in polynomial time.*

In contrast to this positive result, it turns out that maximizing welfare is *impossible* if instead of receiving the explicit functions, we only get oracle access to the valuations. In particular, we show that no deterministic algorithm (even super-polynomial) can find a division approximating the utilitarian or egalitarian optimum by a factor smaller than n. Note that this bound is tight, as every *proportional* division[5] approximates utilitarian and egalitarian welfare by at least n, and many algorithms for finding proportional divisions do exist in the queries model (see, e.g. [RW98] for a survey).

[5]Recall that a division is said to be proportional if it gives each player what she considers to be at least $1/n$ of the total value of the cake.

Theorem 9. *For any $\epsilon > 0$, no deterministic algorithm working in the oracle input model can approximate utilitarian or egalitarian welfare to a factor of $n - \epsilon$, when non-connected pieces are allowed.*

Proof sketch. We discuss utilitarian welfare; the arguments for egalitarian welfare are analoguous. Let A be a deterministic cake division algorithm working in the oracle input model, and fix some $n \in \mathbb{N}$ and $\epsilon > 0$. Consider the operation of the algorithm on the set of preferences in which all players value the entire cake uniformly. In this case, the utilitarian welfare obtained cannot exceed 1. We will now show that for any $\epsilon' > 0$ we can construct a different set of preferences on which the algorithm outputs the same division (with the same welfare), but for which there exists a division achieving utilitarian welfare of $(1 - \epsilon')n$. The theorem will follow by choosing $\epsilon' = \epsilon/n$.

Let $0 = p_0 < p_1 < \ldots < p_{k-1} < p_k = 1$ be the set of (distinct) points that appear in the operation of the algorithm on the input above. I.e. $\{p_i\}_{i=0}^{k}$ is the set of all points a, b for which the algorithm makes a query $v_i(a, b)$ or receives an answer $b = v_i^{-1}(a, x)$, and all the points c in which the algorithm makes cuts in its output division. We create a new instance in which the valuations in the interval between two each consecutive such points (p_j, p_{j+1}) are "rearranged". The value of this interval in the original instance, as well as in the new instance, is $\ell_j = p_{j+1} - p_j$. We divide this interval into $n + 1$ "slivers": the i-th sliver ($1 \leq i \leq n$) is worth $\ell_j - \frac{\epsilon'}{k}$ to player i, and zero to everyone else. The $n + 1$-st sliver of the interval is worth $\frac{\epsilon'}{k}$ for all the players.

It is straightforward to observe that the operation of the algorithm A is identical on the old and new instances, as we constructed the new valuations so that the answer to every query asked in the operation of A is identical in the two instances. This implies that A returns the same division for both instances, and the utilitarian welfare of this division is 1 in both of them. However, in the new instance, any division that allocates every sliver desired only by one player to this player, achieves utilitarian welfare $> (1 - \epsilon')n$. $\square$

6 Open Problems

In this work we have taken the first steps in studying the problem of maximizing welfare in cake cutting with connected pieces. Many interesting problems related to this problem remain open. First and foremost, we believe that it should be possible to obtain a reasonable approximation for the problem of maximizing the egalitarian welfare. (We do have non-trivial algorithms that achieve linear-factor approximations, but we conjecture that better algorithms can be found.) We also conjecture that the approximation ratio for maximizing utilitarian welfare can be improved; it may also be interesting to see if stronger inapproximability results can be shown. Other interesting extensions include:

- *Strategic Behavior:* One implicit assumption in our work was that we have access to the (true) valuations of the players. In reality, the players may have incentive to lie about their valuations. Guo and Conitzer [GC10] and Han et al. [HSTZ11] have considered this problem for a somewhat different setting; the question of what can be achieved truthfully in our setting is still open.

- *2-Dimensional Cake:* The cake cutting literature has generally assumed a one-dimensional cake; indeed, for the purpose of maintaining fairness, which was its main focus, a 2-dimensional cake can be simply "projected" into one dimension, and divided fairly according to the projection. However, this may result in a significant loss of welfare. Therefore, maximizing welfare in allocation of 2-dimensional cakes may require completely different tools and techniques.

Acknowledgment. We thank Zvi Gotthilf for many useful discussions.

References

[AAWY98] Noga Alon, Yossi Azar, Gerhard J. Woeginger, and Tal Yadid. Approximation schemes for scheduling on parallel machines. *Journal of Scheduling*, 1:55–66, 1998.

[AD10] Yonatan Aumann and Yair Dombb. The efficiency of fair division with connected pieces. In *WINE*, pages 26–37, 2010.

[AS07] Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *STOC*, pages 114–121, 2007.

[BCH+12] Xiaohui Bei, Ning Chen, Xia Hua, Biaoshuai Tao, and Endong Yang. Optimal proportional cake cutting with connected pieces. In *AAAI*, 2012.

[BD05] Ivona Bezáková and Varsha Dani. Allocating indivisible goods. *SIGecom Exchanges*, 5(3):11–18, 2005.

[BS06] Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *STOC*, pages 31–40, 2006.

[BT95] Steven J. Brams and Alan D. Taylor. An envy-free cake division protocol. *The American Mathematical Monthly*, 102(1):9–18, 1995.

[BT96] Steven J. Brams and Alan D. Taylor. *Fair Division: From cake cutting to dispute resolution.* Cambridge University Press, New York, NY, USA, 1996.

[CCK09] Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. On allocating goods to maximize fairness. In *FOCS*, pages 107–116, 2009.

[CKKK09] Ioannis Caragiannis, Christos Kaklamanis, Panagiotis Kanellopoulos, and Maria Kyropoulou. The efficiency of fair division. In *WINE*, pages 475–482, 2009.

[CLPP10] Yiling Chen, John Lai, David C. Parkes, and Ariel D. Procaccia. Truth, justice, and cake cutting. In *AAAI*, 2010.

[CLPP11] Yuga J. Cohler, John Lai, David C. Parkes, and Ariel Procaccia. Optimal envy-free cake cutting. In *AAAI-11: Proc. 25th AAAI Conference on Artificial Intelligence*, 2011.

[DF99] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.

[DS61] L. E. Dubins and E. H. Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1):1–17, Jan 1961.

[EP84] S. Even and A. Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285 – 296, 1984.

[Fei08] Uriel Feige. On allocations that maximize fairness. In *SODA*, pages 287–293, 2008.

[GC10] Mingyu Guo and Vincent Conitzer. Strategy-proof allocation of multiple items between two agents without payments or priors. In *AAMAS*, pages 881–888, 2010.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman, 1979.

[HSTZ11] Li Han, Chunzhi Su, Linpeng Tang, and Hongyang Zhang. On strategy-proof allocation without payments or priors. In *WINE*, pages 182–193, 2011.

[LMMS04] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *ACM Conference on Electronic Commerce*, pages 125–131, 2004.

[LST90] Jan Karel Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.*, 46:259–271, 1990.

[Pro09] Ariel D. Procaccia. Thou shalt covet thy neighbor's cake. In *IJCAI*, pages 239–244, 2009.

[RW98] Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can.* A K Peters, Ltd., Natick, MA, USA, 1998.

[Ste49] H. Steinhaus. Sur la division pragmatique. *Econometrica*, 17(Supplement: Report of the Washington Meeting):315–319, Jul 1949.

[Str80] Walter Stromquist. How to cut a cake fairly. *The American Mathematical Monthly*, 87(8):640–644, 1980.

[Str08] Walter Stromquist. Envy-free cake divisions cannot be found by finite protocols. *Electronic Journal of Combinatorics*, 15(1), Jan 2008.

[SW03] Jiri Sgall and Gerhard J. Woeginger. A lower bound for cake cutting. In *ESA*, pages 459–469, 2003.

[Woe97] Gerhard J. Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149 – 154, 1997.

[Ziv11] Roie Zivan. Can trust increase the efficiency of cake cutting algorithms? In *AAMAS*, pages 1145–1146, 2011.

Yonatan Aumann
Department of Computer Science
Bar-Ilan University
Ramat Gan 52900, Israel
Email: `aumann@cs.biu.ac.il`

Yair Dombb
Department of Computer Science
Bar-Ilan University
Ramat Gan 52900, Israel
Email: `yair.biu@gmail.com`

Avinatan Hassidim
Department of Computer Science
Bar-Ilan University
Ramat Gan 52900, Israel
Email: `avinatanh@gmail.com`

Possible and Necessary Winners of Partial Tournaments*

Haris Aziz, Markus Brill, Felix Fischer,
Paul Harrenstein, Jérôme Lang, and Hans Georg Seedig

Abstract

We study the problem of computing possible and necessary winners for partially specified weighted and unweighted tournaments. This problem arises naturally in elections with incompletely specified votes, partially completed sports competitions, and more generally in any scenario where the outcome of some pairwise comparisons is not yet fully known. We specifically consider a number of well-known solution concepts—including the uncovered set, Borda, ranked pairs, and maximin—and show that for most of them possible and necessary winners can be identified in polynomial time. These positive algorithmic results stand in sharp contrast to earlier results concerning possible and necessary winners given partially specified preference profiles.

1 Introduction

Many multi-agent situations can be modeled and analyzed using weighted or unweighted tournaments. Prime examples are voting scenarios in which pairwise comparisons between alternatives are decided by majority rule and sports competitions that are organized as round-robin tournaments. Other application areas include webpage and journal ranking, biology, psychology, and AI (also see [6], and the references therein). More generally, tournaments and tournament solutions are used as a mathematical tool for the analysis of all kinds of situations where a choice among a set of alternatives has to be made exclusively on the basis of pairwise comparisons.

When choosing from a tournament, relevant information may only be partly available. This could be because some preferences are yet to be elicited, some matches yet to be played, or certain comparisons yet to be made. In such cases, it is natural to speculate which are the potential and inevitable outcomes on the basis of the information already at hand.

For complete tournaments, a number of attractive solution concepts have been proposed (see, e.g., [6, 17]). Given any such solution concept S, *possible winners* of a partial tournament G are defined as alternatives that are selected by S in *some* completion of G, and *necessary winners* are alternatives that are selected in *all* completions. By a completion we here understand a complete tournament extending G.

In this paper we address the computational complexity of identifying the possible and necessary winners for a number of solution concepts whose winner determination problem for complete tournaments is tractable. We consider four of the most common tournament solutions—namely, Condorcet winners ($COND$), the Copeland solution (CO), the top cycle (TC), and the uncovered set (UC)—and three common solutions for weighted tournaments—Borda (BO), maximin (MM) and ranked pairs (RP). For each of these solution concepts, we characterize the complexity of the following problems: deciding whether a given alternative is a possible winner (PW), deciding whether a given alternative is a necessary winner (NW), and deciding whether a given subset of alternatives equals the set

*A previous version of this paper has been accepted at AAMAS-2012. New results include speeding up pseudo-polynomial time algorithms to strongly polynomial time for PWS_{BO} (Thm. 8) and PWS_{MM} (Thm. 12).

S	PW_S		NW_S		PWS_S	
COND	in P	[16]	in P	[16]	in P	(Thm. 1)
CO	in P	(Thm. 2)[a]	in P	(Thm. 2)[a]	in P	(Thm. 2)
TC	in P	[16][a]	in P	[16]	in P	(Thm. 3)
UC	in P	(Thm. 4)	in P	(Thm. 5)	NP-C	(Thm. 6)
BO	in P	(Thm. 7)[a]	in P	(Thm. 9)	in P	(Thm. 8)
MM	in P	(Thm. 10)[a]	in P	(Thm. 11)	in P	(Thm. 12)
RP	NP-C	(Thm. 13)	coNP-C	(Thm. 14)	NP-C	(Cor. 1)

[a] This P-time result contrasts with the intractability of the same problem for partial preference profiles [16, 25].

Table 1: Complexity of computing possible winners (PW) and necessary winners (NW) and of checking whether a given subset of alternatives is a possible winning set (PWS) under different solution concepts given partial tournaments.

of winners in some completion (PWS). These problems can be challenging, as even unweighted partial tournaments may allow for an exponential number of completions. Our results are encouraging, in the sense that most of the problems can be solved in polynomial time. Table 1 summarizes our findings.

Similar problems have been considered before. For Condorcet winners, voting trees and the top cycle, it was already shown that possible and necessary winners are computable in polynomial time [16, 19, 20]. The same holds for computing possible Copeland winners that were considered in the context of sports tournaments [8].

A more specific setting that is frequently considered within the area of computational social choice differs from our setting in a subtle but important way that is worth being pointed out. There, tournaments are assumed to arise from pairwise majority comparisons on the basis of a profile of individual voters' preferences.[1] Since a *partial* preference profile R need not conclusively settle every majority comparison, it may give rise to a *partial* tournament only. There are two natural ways to define possible and necessary winners for a partial preference profile R and solution concept S. The first is to consider the completions of the incomplete tournament $G(R)$ corresponding to R and the winners under S in these. This is covered by our more general setting. The second is to consider the completions of R and the winners under S in the corresponding tournaments.[2] Since every tournament corresponding to a completion of R is also a completion of $G(R)$ but not necessarily the other way round, the second definition gives rise to a *stronger* notion of a possible winner and a *weaker* notion of a necessary winner. Interestingly, and in sharp contrast to our results, determining these stronger possible and weaker necessary winners is computationally hard for many voting rules [16, 25].

In the context of this paper, we do not assume that tournaments arise from majority comparisons in voting or from any other specific procedure. This approach has a number of advantages. Firstly, it matches the diversity of settings to which tournament solutions are applicable, which goes well beyond social choice and voting. For instance, our results also apply to a question commonly encountered in sports competitions, namely, which teams can still win the cup and which future results this depends on (see, e.g., [8, 14]). Secondly, (partial) tournaments provide an informationally sustainable way of representing the relevant aspects of many situations while maintaining a workable level of abstraction and concise-

[1] See, e.g., [1, 2, 15, 24, 25] for the basic setting, [3] for parameterized complexity results, [12, 13] for probabilistic settings, and [7, 26] for settings with a variable set of alternatives.

[2] These two ways of defining possible and necessary winners are compared (both theoretically and experimentally) in [16, 20] for three solution concepts: Condorcet winners, voting trees and the top cycle.

ness. For instance, in the social choice setting described above, the partial tournament induced by a partial preference profile is a much more succinct piece of information than the preference profile itself. Finally, specific settings may impose restrictions on the feasible extensions of partial tournaments. The positive algorithmic results in this paper can be used to efficiently approximate the sets of possible and necessary winners in such settings, where the corresponding problems may be intractable. The voting setting discussed above serves to illustrate this point.

2 Preliminaries

A *partial tournament* is a pair $G = (V, E)$ where V is a finite set of alternatives and $E \subseteq V \times V$ an asymmetric relation on V, i.e., $(x, y) \in E$ implies $(y, x) \notin E$. If $(x, y) \in E$ we say that x *dominates* y. A *(complete) tournament* T is a partial tournament (V, E) for which E is also complete, i.e., either $(x, y) \in E$ or $(y, x) \in E$ for all distinct $x, y \in V$. We denote the class of complete tournaments by $\mathscr{T}$.

Let $G = (V, E)$ be a partial tournament. Another partial tournament $G' = (V', E')$ is called an *extension* of G, denoted $G \leq G'$, if $V = V'$ and $E \subseteq E'$. If E' is complete, G' is called a *completion of* G. We write $[G]$ for the set of completions of G, i.e., $[G] = \{T \in \mathscr{T} : G \leq T\}$.

For each $x \in V$, we define the *dominion of* x *in* G by $D^+_G(x) = \{y \in V : (x, y) \in E\}$, and the *dominators of* x *in* G by $D^-_G(x) = \{y \in V : (y, x) \in E\}$. For $X \subseteq V$, we let $D^+_G(X) = \bigcup_{x \in X} D^+_G(x)$ and $D^-_G(X) = \bigcup_{x \in X} D^-_G(x)$.

For given $G = (V, E)$ and $X \subseteq V$, we further write $E^{X\rightarrow}$ for the set of edges obtained from E by adding all missing edges from alternatives in X to alternatives not in X, i.e.,

$$E^{X\rightarrow} = E \cup \{(x, y) \in X \times V : y \notin X \text{ and } (y, x) \notin E\}.$$

We use $E^{X\leftarrow}$ as an abbreviation for $E^{V\setminus X\rightarrow}$, and respectively write $E^{x\rightarrow}$, $E^{x\leftarrow}$, $G^{X\rightarrow}$, and $G^{X\leftarrow}$ for $E^{\{x\}\rightarrow}$, $E^{\{x\}\leftarrow}$, $(V, E^{X\rightarrow})$, and $(V, E^{X\leftarrow})$.

Let n be a positive integer. A *partial n-weighted tournament* is a pair $G = (V, w)$ consisting of a finite set of alternatives V and a weight function $w : V \times V \to \{0, \ldots, n\}$ such that for each pair $(x, y) \in V \times V$ with $x \neq y$, $w(x, y) + w(y, x) \leq n$. We say that $T = (V, w)$ is a *(complete) n-weighted tournament* if for all $x, y \in V$ with $x \neq y$, $w(x, y) + w(y, x) = n$. A (partial or complete) *weighted* tournament is a (partial or complete) n-weighted tournament for some $n \in \mathbb{N}$. The class of n-weighted tournaments is denoted by $\mathscr{T}_n$. Observe that with each partial 1-weighted tournament (V, w) we can associate a partial tournament (V, E) by setting $E = \{(x, y) \in V : w(x, y) = 1\}$. Thus, (partial) n-weighted tournaments can be seen to generalize (partial) tournaments, and we may identify $\mathscr{T}_1$ with $\mathscr{T}$.

The notations $G \leq G'$ and $[G]$ can be extended naturally to partial n-weighted tournaments $G = (V, w)$ and $G' = (V', w')$ by letting $(V, w) \leq (V', w')$ if $V = V'$ and $w(x, y) \leq w'(x, y)$ for all $x, y \in V$, and $[G] = \{T \in \mathscr{T}_n : G \leq T\}$.

For given $G = (V, w)$ and $X \subseteq V$, we further define $w^{X\rightarrow}$ such that for all $x, y \in V$,

$$w^{X\rightarrow}(x, y) = \begin{cases} n - w(y, x) & \text{if } x \in X \text{ and } y \notin X, \\ w(x, y) & \text{otherwise,} \end{cases}$$

and set $w^{X\leftarrow} = w^{V\setminus X\rightarrow}$. Moreover, $w^{x\rightarrow}$, $w^{x\leftarrow}$, $G^{X\rightarrow}$, and $G^{X\leftarrow}$ are defined in the obvious way.

We use the term *solution concept* for functions S that associate with each (complete) tournament $T = (V, E)$, or with each (complete) weighted tournament $T = (V, w)$, a choice set $S(T) \subseteq V$. A solution concept S is called *resolute* if $|S(T)| = 1$ for each tournament T.

In this paper we will consider the following solution concepts: *Condorcet winners* ($COND$), *Copeland* (CO), *top cycle* (TC), and *uncovered set* (UC) for tournaments, and *maximin* (MM), *Borda* (BO), and *ranked pairs* (RP) for weighted tournaments. Of these only ranked pairs is resolute. Formal definitions will be provided later in the paper.

3 Possible & Necessary Winners

A solution concept selects alternatives from complete tournaments or complete weighted tournaments. A partial (weighted) tournament, on the other hand, can be extended to a number of complete (weighted) tournaments, and a solution concept selects a (potentially different) set of alternatives for each of them.

For a given solution concept S, we can thus define the set of *possible winners* for a partial (weighted) tournament G as the set of alternatives selected by S from *some* completion of G, i.e., as $PW_S(G) = \bigcup_{T\in[G]} S(T)$. Analogously, the set of *necessary winners* of G is the set of alternatives selected by S from *every* completion of G, i.e., $NW_S(G) = \bigcap_{T\in[G]} S(T)$. We can finally write $PWS_S(G) = \{S(T) : T \in [G]\}$ for the set of *sets* of alternatives that S selects for the different completions of G.

Note that $NW_S(G)$ may be empty even if S selects a non-empty set of alternatives for each tournament $T \in [G]$, and that $|PWS_S(G)|$ may be exponential in the number of alternatives of G. It is also easily verified that $G \leq G'$ implies $PW_S(G') \subseteq PW_S(G)$ and $NW_S(G) \subseteq NW_S(G')$, and that $PW_S(G) = \bigcup_{G\leq G'} NW_S(G')$ and $NW_S(G) = \bigcap_{G\leq G'} PW_S(G')$.

Deciding membership in the sets $PW_S(G)$, $NW_S(G)$, and $PWS_S(G)$ for a given solution concept S and a partial (weighted) tournament G is a natural computational problem. We will respectively refer to these problems as PW_S, NW_S, and PWS_S, and will study them for the solution concepts mentioned at the end of the previous section.[3]

For complete tournaments T we have $[T] = \{T\}$ and thus $PW_S(T) = NW_S(T) = S(T)$ and $PWS_S(T) = \{S(T)\}$. As a consequence, for solution concepts S with an NP-hard winner determination problem—like *Banks*, *Slater*, and *TEQ*—the problems PW_S, NW_S, and PWS_S are NP-hard as well. We therefore restrict our attention to solution concepts for which winners can be computed in polynomial time.

For irresolute solution concepts, PWS_S may appear a more complex problem than PW_S. We are, however, not aware of a polynomial-time reduction from PW_S to PWS_S. The relationship between these problems may also be of interest for the "classic" possible winner setting with partial preference profiles.

4 Unweighted tournaments

In this section, we consider the following well-known solution concepts for unweighted tournaments: Condorcet winners, Copeland, top cycle, and uncovered set. Weighted tournaments will then be considered in Section 5.

4.1 Condorcet Winners

Condorcet winners are a very simple solution concept and will provide a nice warm-up. An alternative $x \in V$ is a *Condorcet winner* of a complete tournament $T = (V, E)$ if it dominates all other alternatives, i.e., if $(x, y) \in E$ for all $y \in V \setminus \{x\}$. The set of Condorcet winners of tournament T will be denoted by $COND(T)$; obviously this set is always either a singleton or empty.

[3] Formally, the input for each of the problems consists of an encoding of the partial (n-weighted) tournament G and, for partial n-weighted tournaments, the number n.

It is readily appreciated that the possible Condorcet winners of a partial tournament $G = (V, E)$ are precisely the undominated alternatives, and that a necessary Condorcet winner of G should already dominate all other alternatives. Both properties can be verified in polynomial time.

Each of the sets in $PWS_{COND}(G)$ is either a singleton or the empty set, and determining membership for a singleton is obviously tractable. Checking whether $\emptyset \in PWS_{COND}(G)$ is not quite that simple. First observe that $\emptyset \in PWS_{COND}(G)$ if and only if there is an extension G' of G in which every alternative is dominated by some other alternative. Given a particular $G = (V, E)$, we can define an extension $G' = (V, E')$ of G by iteratively adding edges from dominated alternatives to undominated ones until this is no longer possible. Formally, let

$$E_0 = E \text{ and } E_{i+1} = E_i \cup \{(x, y) \in X_i \times Y_i : (y, x) \notin E_i\},$$

where X_i and Y_i denote the dominated and undominated alternatives of (V, E_i), respectively. Finally define $E' = \bigcup_{i=0}^{|V|} E_i$, and observe that this set can be computed in polynomial time.

Now, for every undominated alternative x of G' and every dominated alternative y of G', we not only have $(x, y) \in E'$, but also $(x, y) \in E$. This is the case because in the inductive definition of E' only edges from dominated to undominated alternatives are added in every step. It is therefore easily verified that $PWS_{COND}(G)$ contains $\emptyset$ if and only if the set of undominated alternatives in G' is either empty or is of size three or more. We have shown the following easy result.

Theorem 1. *PW_{COND}, NW_{COND}, and PWS_{COND} can be solved in polynomial time.*

The results for PW_{COND} and NW_{COND} also follow from Proposition 2 of Lang et al. [16] and Corollary 2 of Konczak and Lang [15]. We further note that Theorem 1 is a corollary of corresponding results for maximin in Section 5.2. The reason is that a Condorcet winner is the maximin winner of a 1-weighted tournament, and a tournament does not admit a Condorcet winner if and only if all alternatives are maximin winners.

4.2 Copeland

Copeland's solution selects alternatives based on the number of other alternatives they dominate. Define the *Copeland score* of an alternative x in tournament $T = (V, E)$ as $s_{CO}(x, T) = |D^+_T(x)|$. The set $CO(T)$ then consists of all alternatives that have maximal Copeland score. Since Copeland scores coincide with Borda scores in the case of 1-weighted tournaments, the following is a direct corollary of the results in Section 5.1.

Theorem 2. *NW_{CO}, PW_{CO}, and PWS_{CO} can be solved in polynomial time.*

PW_{CO} can alternatively be solved via a polynomial-time reduction to maximum network flow (see, e.g., [8], p. 51).

4.3 Top Cycle

A subset $X \subseteq V$ of alternatives in a (partial or complete) tournament (V, E) is *dominant* if every alternative in X dominates every alternative outside X. The *top cycle* of a tournament $T = (V, E)$, denoted by $TC(T)$, is the unique *minimal* dominant subset of V.

Lang et al. have shown that possible and necessary winners for TC can be computed efficiently by greedy algorithms ([16], Corollaries 1 and 2). For PWS_{TC}, we not only have to check that there exists a completion such that the set in question is dominating, but also that there is no smaller dominating set. It turns out that this can still be done in polynomial time.

Theorem 3. *PWS_{TC} can be solved in polynomial time.*

Proof sketch. Consider a partial tournament $G = (V, E)$ and a set $X \subseteq V$ of alternatives. If X is a singleton, the problem reduces to checking whether $X \in PWS_{COND}(G)$. If X is of size two or if one of its elements is dominated by an outside alternative, $X \notin PWS_{TC}(G)$. Therefore, we can without loss of generality assume that $|X| \geq 3$ and $(y, x) \notin E$ for all $y \in V \setminus X$ and $x \in X$. The *Smith set* of a partial tournament is defined as the minimal dominant subset of alternatives [22].[4] It can be shown that there exists a completion $T \in [G]$ with $TC(T) = X$ if and only if the Smith set of the partial tournament $(X, E|_{X \times X})$ equals the whole set X. Since Brandt et al. [4] have shown that the Smith set of a partial tournament can be computed efficiently, the theorem follows. □

4.4 Uncovered Set

Given a tournament $T = (V, E)$, an alternative $x \in V$ is said to *cover* another alternative $y \in V$ if $D_T^+(y) \subseteq D_T^+(x)$, i.e., if every alternative dominated by y is also dominated by x. The *uncovered set* of T, denoted by $UC(T)$, then is the set of alternatives that are not covered by some other alternative. A useful alternative characterization of the uncovered set is via the *two-step principle*: an alternative is in the uncovered set if and only if it can reach every other alternative in at most two steps.[5] Formally, $x \in UC(T)$ if and only if for all $y \in V \setminus \{x\}$, either $(x, y) \in E$ or there is some $z \in V$ with $(x, z), (z, y) \in E$. We denote the two-step dominion $D_E^+(D_E^+(x))$ of an alternative x by $D_E^{++}(x)$.

We first consider PW_{UC}, for which we check for each alternative whether it can be reinforced to reach every other alternative in at most two steps.

Theorem 4. *PW_{UC} can be solved in polynomial time.*

Proof. For a given partial tournament $G = (V, E)$ and an alternative $x \in V$, we check whether x is in $UC(T)$ for some completion $T \in [G]$.

Consider the graph $G' = (V, E'')$ where E'' is derived from E as follows. First, we let $D^+(x)$ grow as much as possible by letting $E' = E^{x\rightarrow}$. Then, we do the same for its two-step dominion by defining E'' as $E'^{D_{E'}^+(x)\rightarrow}$. Now it can be shown that $x \in PW_{UC}(G)$ if and only if $V = \{x\} \cup D_{E''}^+(x) \cup D_{E''}^{++}(x)$. □

A similar argument yields the following.

Theorem 5. *NW_{UC} can be solved in polynomial time.*

Proof. For a given partial tournament $G = (V, E)$ and an alternative $x \in V$, we check whether x is in $UC(T)$ for all completions $T \in [G]$.

Consider the graph $G' = (V, E'')$ with E'' defined as follows. First, let $E' = E^{x\leftarrow}$. Then, expand it to $E'' = E'^{D_{E'}^-(x)\rightarrow}$. Intuitively, this makes it as hard as possible for x to beat alternatives outside of its dominion in two steps. Then it can be shown that $x \in NW_{UC}(G)$ if and only if $V = \{x\} \cup D_{E''}^+(x) \cup D_{E''}^{++}(x)$. □

For all solution concepts considered so far—Condorcet winners, Copeland, and top cycle—PW and PWS have the same complexity. One might wonder whether a result like this holds more generally, and whether there could be a polynomial-time reduction from PWS to PW. The following result shows that this is not the case, unless P=NP.

Theorem 6. *PWS_{UC} is NP-complete.*

This can be shown by a rather intricate reduction from Sat. We have to omit the construction due to space constraints but a sketch is presented in the appendix.

[4] For complete tournaments, the Smith set coincides with the top cycle.
[5] In graph theory, vertices satisfying this property are often called *kings*.

5 Weighted Tournaments

We now turn to weighted tournaments, and in particular consider the solution concepts Borda, maximin, and ranked pairs.

5.1 Borda

The Borda solution (BO) is typically used in a voting context, where it is construed as based on voters' rankings of the alternatives: each alternative receives $|V|-1$ points for each time it is ranked first, $|V|-2$ points for each time it is ranked second, and so forth; the solution concept then chooses the alternatives with the highest total number of points. In the more general setting of weighted tournaments, the *Borda score* of alternative $x \in V$ in $G=(V,w)$ is defined as $s_{BO}(x,G)=\sum_{y\in V\setminus\{x\}} w(x,y)$ and the *Borda winners* are the alternatives with the highest Borda score. If $w(x,y)$ represents the number of voters that rank x higher than y, the two definitions are equivalent.

Before we proceed further, we define the notion of a b-matching, which will be used in the proofs of two of our results. Let $H=(V_H,E_H)$ be an undirected graph with vertex capacities $b: V_H \to \mathbb{N}_0$. Then, a *b-matching* of H is a function $m: E_H \to \mathbb{N}_0$ such that for all $v \in V_H$, $\sum_{e\in\{e'\in E_H : v\in e'\}} m(e) \le b(v)$. The *size* of b-matching m is defined as $\sum_{e\in E_H} m(e)$. It is easy to see that if $b(v)=1$ for all $v\in V_H$, then a maximum size b-matching is equivalent to a maximum cardinality matching. In a b-matching problem with upper *and* lower bounds, there further is a function $a: V_H \to \mathbb{N}_0$. A feasible b-matching then is a function $m: E_H \to \mathbb{N}_0$ such that $a(v) \le \sum_{e\in\{e'\in E_H : v\in e'\}} m(e) \le b(v)$.

If H is bipartite, then the problem of computing a maximum size feasible b-matching with lower and upper bounds can be solved in strongly polynomial time ([21], Chapter 21). We will use this fact to show that PW_{BO} and PWS_{BO} can both be solved in polynomial time. While the following result for PW_{BO} can be shown using Theorem 6.1 of [14], we give a direct proof that can then be extended to PWS_{BO}.

Theorem 7. *PW_{BO} can be solved in polynomial time.*

Proof sketch. Let $G=(V,w)$ be a partial n-weighted tournament, $x\in V$. We give a polynomial-time algorithm for checking whether $x \in PW_{BO}(G)$, via a reduction to the problem of computing a maximum size b-matching of a bipartite graph.

Let $G^{x\to}=(V,w^{x\to})$ denote the graph obtained from G by maximally reinforcing x, and $s^*=s_{BO}(x,G^{x\to})$ the Borda score of x in $G^{x\to}$. From $G^{x\to}$, we then construct a bipartite graph $H=(V_H,E_H)$ with vertices $V_H = V\setminus\{x\}\cup E^{<n}$, where $E^{<n}=\{\{i,j\}\subseteq V\setminus\{x\} : w(i,j)+w(j,i)<n\}$,[6] and edges $E_H=\{\{v,e\} : v\in V\setminus\{x\} \text{ and } v\in e\in E^{<n}\}$. We further define vertex capacities $b: V_H\to\mathbb{N}_0$ such that $b(\{i,j\})=n-w(i,j)-w(j,i)$ for $\{i,j\}\in E^{<n}$ and $b(v)=s^*-s_{BO}(v,G^{x\to})$ for $v\in V\setminus\{x\}$.

Now observe that in any completion $T=(V,w')\in[G^{x\to}]$, $w'(i,j)+w'(j,i)=n$ for all $i,j\in V$ with $i\neq j$. The sum of the Borda scores in T is therefore $n|V|(|V|-1)/2$. Some of the weight has already been used up in $G^{x\to}$; the weight which has not yet been used up is equal to $\alpha = n|V|(|V|-1)/2 - \sum_{v\in V} s_{BO}(v,G^{x\to})$. We claim that $x\in PW_{BO}(G)$ if and only if H has a b-matching of size at least α.

□

This idea can be extended to a polynomial-time algorithm for PWS_{BO} where we use a similar construction for a given $G=(V,w)$, a candidate set $X\subset X$ and a target Borda score s^*. Binary search can be used to efficiently search the interval $I=[\max_{x\in X} s_{BO}(x,G), n(|V|-1)]$ of possible target scores. The full proof is omitted.

[6] Note that $w(i,j)=w^{x\to}(i,j)$ for alternatives $i,j\in V\setminus\{x\}$.

Theorem 8. *PWS_{BO} can be solved in polynomial time.*

We conclude this section by showing that NW_{BO} can be solved in polynomial time as well.

Theorem 9. *NW_{BO} can be solved in polynomial time.*

Proof. Let $G = (V, w)$ be a partial weighted tournament, $x \in V$. We give a polynomial-time algorithm for checking whether $x \in NW_{BO}(G)$.

Let $G' = G^{x\leftarrow}$. We want to check whether some other alternative $y \in V \setminus \{x\}$ can achieve a Borda score of more than $s^* = s_{BO}(x, G')$. This can be done separately for each $y \in V \setminus \{x\}$ by reinforcing it as much as possible in G'. If for some y, $s_{BO}(y, G'^{y\rightarrow}) > s^*$, then $x \notin NW_{BO}(G)$. If, on the other hand, $s_{BO}(y, G'^{y\rightarrow}) \leq s^*$ for all $y \in V \setminus \{x\}$, then $x \in NW_{BO}(G)$. □

Since the Borda and Copeland solutions coincide in unweighted tournaments, the above results imply that PW_{CO} and NW_{CO} can be solved in polynomial time. The same is true for PWS_{CO}, because the Copeland score is bounded by $|V| - 1$.

5.2 Maximin

The *maximin score* $s_{MM}(x, T)$ of an alternative x in a weighted tournament $T = (V, w)$, is given by its worst pairwise comparison, i.e., $s_{MM}(x, T) = \min_{y \in V \setminus \{x\}} w(x, y)$. The *maximin solution*, also known as *Simpson's method* and denoted by MM, returns the set of all alternatives with the highest maximin score.

We first show that PW_{MM} is polynomial-time solvable by reducing it to the problem of finding a maximum cardinality matching of a graph.

Theorem 10. *PW_{MM} can be solved in polynomial time.*

Proof sketch. We show how to check whether $x \in PW_{MM}(G)$ for a partial n-weighted tournament $G = (V, w)$. Consider the graph $G^{x\rightarrow} = (V, w^{x\rightarrow})$. Then, $s_{MM}(x, G^{x\rightarrow})$ is the best possible maximin score x can get among all completions of G. If $s_{MM}(x, G^{x\rightarrow}) \geq \frac{n}{2}$, then we have $s_{MM}(y, T) \leq w^{x\rightarrow}(y, x) \leq \frac{n}{2}$ for every $y \in V \setminus \{x\}$ and every completion $T \in [G^{x\rightarrow}]$ and therefore $x \in PW_{MM}(G)$. Now consider $s_{MM}(x, G^{x\rightarrow}) < \frac{n}{2}$. We will reduce the problem of checking whether $x \in PW_{MM}(G)$ to that of finding a maximum cardinality matching, which is known to be solvable in polynomial time [11]. We want to find a completion $T \in [G^{x\rightarrow}]$ such that $s_{MM}(x, T) \geq s_{MM}(y, T)$ for all $y \in V \setminus \{x\}$. If there exists a $y \in V \setminus \{x\}$ such that $s_{MM}(x, G^{x\rightarrow}) < s_{MM}(y, G^{x\rightarrow})$, then we already know that $x \notin PW_{MM}(G)$. Otherwise, each $y \in V \setminus \{x\}$ derives its maximin score from at least one particular edge (y, z) where $z \in V \setminus \{x, y\}$ and $w(y, z) \leq s_{MM}(x, G^{x\rightarrow})$. Moreover, it is clear that in any completion, y and z cannot both achieve a maximin score of less than $s_{MM}(x, G^{x\rightarrow})$ from edges (y, z) and (z, y) at the same time.

Construct the following undirected and unweighted graph $H = (V_H, E_H)$ where $V_H = V \setminus \{x\} \cup \{\{i, j\} \subseteq V : i \neq j\}$. Build up E_H such that: $\{i, \{i, j\}\} \in E_H$ if and only if $i \neq j$ and $w^{x\rightarrow}(i, j) \leq s_{MM}(x, G^{x\rightarrow})$. In this way, if i is matched to $\{i, j\}$ in H, then i derives a maximin score of less than or equal to $s_{MM}(x, G^{x\rightarrow})$ from his comparison with j. Clearly, H is polynomial in the size of G. Then, the claim is that $x \in PW_{MM}(G)$ if and only if there exists a matching of cardinality $|V| - 1$ in H. □

For NW_{MM} we apply a similar technique as for NW_{BO}: to see whether $x \in NW_{MM}(G)$, we start from the graph $G^{x\leftarrow}$ and check whether some other alternative can achieve a higher maximin score than x in a completion of $G^{x\leftarrow}$.

Theorem 11. *NW_{MM} can be solved in polynomial time.*

We conclude the section by showing that PWS_{MM} can be solved in polynomial time. The proof proceeds by identifying the maximin values that could potentially be achieved simultaneously by all elements of the set in question, and solving the problem for each of these values using similar techniques as in the proof of Theorem 10. Only a polynomially bounded number of problems need to be considered.

Theorem 12. *PWS_{MM} can be solved in polynomial time.*

5.3 Ranked Pairs

The method of *ranked pairs* (*RP*) is the only resolute solution concept considered in this paper. Given a weighted tournament $T = (V, w)$, it returns the unique undominated alternative of a transitive tournament T' on V constructed in the following manner. First order the (directed) edges of T in decreasing order of weight, breaking ties according to some exogenously given tie-breaking rule. Then consider the edges one by one according to this ordering. If the current edge can be added to T' without creating a cycle, then do so; otherwise discard the edge.[7]

It is readily appreciated that this procedure, and thus the winner determination problem for RP, is computationally tractable. The possible winner problem, on the other hand, turns out to be NP-hard. This also shows that tractability of the winner determination problem, while necessary for tractability of PW, is not generally sufficient.

Theorem 13. *PW_{RP} is NP-complete.*

Proof sketch. Membership in NP is obvious, as for a given completion and a given tie-breaking rule, the ranked pairs winner can be found efficiently.

NP-hardness can be shown by a reduction from SAT. For a Boolean formula φ in conjunctive normal-form with a set C of clauses and set P of propositional variables, we construct a partial 8-weighted tournament $G_\varphi = (V_\varphi, w_\varphi)$ as follows. For each variable $p \in P$, V_φ contains two *literal alternatives* p and $\bar{p}$ and two *auxiliary alternatives* p' and $\bar{p}'$. For each clause $c \in C$, there is an alternative c. Finally, there is an alternative d for which membership in $PW_{RP}(G_\varphi)$ is to be decided.

In order to conveniently describe the weight function w_φ, let us introduce the following terminology. For two alternatives $x, y \in V_\varphi$, say that there is a *heavy* edge from x to y if $w_\varphi(x, y) = 8$ (and therefore $w_\varphi(y, x) = 0$). A *medium* edge from x to y means $w_\varphi(x, y) = 6$ and $w_\varphi(y, x) = 2$, and a *light* edge from x to y means $w_\varphi(x, y) = 5$ and $w_\varphi(y, x) = 3$. Finally, a *partial* edge between x and y means $w_\varphi(x, y) = w_\varphi(y, x) = 1$.

We are now ready to define w_φ. For each variable $p \in P$, we have heavy edges from p to $\bar{p}'$ and from $\bar{p}$ to p', and partial edges between p and p' and between $\bar{p}$ and $\bar{p}'$. For each clause $c \in C$, we have a medium edge from c to d and a heavy edge from the literal alternative $\ell_i \in \{p, \bar{p}\}$ to c if the corresponding literal ℓ_i appears in the clause c. Finally, we have heavy edges from d to all auxiliary alternatives and light edges from d to all literal alternatives. For all pairs x, y for which no edge has been specified, we define $w_\varphi(x, y) = w_\varphi(y, x) = 4$.

Observe that the only pairs of alternatives for which w_φ is not fully specified are those pairs that are connected by a partial edge. It can be shown that alternative d is a possible

[7]The variant of ranked pairs originally proposed by Tideman [23], which was also used by Xia and Conitzer [25], instead chooses a set of alternatives, containing any alternative that is selected by the above procedure for *some* way of breaking ties among edges with equal weight. We do not consider this irresolute version of ranked pairs because it was recently shown that winner determination for this variant is NP-hard [5]. As mentioned in Section 3, this immediately implies that all problems concerning possible or necessary winners are NP-hard as well.

ranked pairs winner in G_φ if and only if φ is satisfiable. Intuitively, choosing a completion w' of w_φ such that $w'(p', p)$ is large and $w'(\bar{p}', \bar{p})$ is small corresponds to setting the variable p to "true." □

Since the ranked pairs method is resolute, hardness of PWS_{RP} follows immediately.

Corollary 1. *PWS_{RP} is NP-complete.*

Computing necessary ranked pairs winners turns out to be coNP-complete. This is again somewhat surprising, as computing necessary winners is often considerably easier than computing possible winners, both for partial tournaments and partial preference profiles [25].

Theorem 14. *NW_{RP} is coNP-complete.*

Proof sketch. Membership in coNP is again obvious. For hardness, we give a reduction from UnSat that is a slight variation of the reduction in the proof of Theorem 13. We introduce a new alternative d^*, which has heavy edges to all alternatives in V_φ except d. Furthermore, there is a light edge from d to d^*. It can be shown that d^* is a necessary ranked pairs winner in this partial 8-weighted tournament if and only if φ is unsatisfiable. □

6 Discussion

The problem of computing possible and necessary winners for partial preference profiles has recently received a lot of attention. In this paper, we have investigated this problem in a setting where partially specified (weighted or unweighted) *tournaments* instead of profiles are given as input. We have summarized our findings in Table 1.

A key conclusion is that computational problems for partial tournaments can be significantly easier than their counterparts for partial profiles. For example, possible Borda or maximin winners can be found efficiently for partial tournaments, whereas the corresponding problems for partial profiles are NP-complete [25].

While tractability of the winner determination problem is necessary for tractability of the possible or necessary winners problems, the results for ranked pairs in Section 5.3 show that it is not sufficient. We further considered the problem of deciding whether a given subset of alternatives equals the winner set for some completion of the partial tournament. The results for the uncovered set in Section 4.4 imply that this problem cannot be reduced to the computation of possible or necessary winners, but whether a reduction exists in the opposite direction remains an open problem.

Partial tournaments have also been studied in their own right, independent of their possible completions. For instance, Peris and Subiza [18] and Dutta and Laslier [10] have generalized several tournament solutions to incomplete tournaments by directly adapting their definitions. In this context, the notion of possible winners suggests a canonical way to generalize a tournament solution to incomplete tournaments. The positive computational results in this paper are an indication that this may be a promising approach.

Furthermore, we have not examined the complexity of computing possible and necessary winners for some attractive tournament solutions such as the minimal covering set, the bipartisan set [17] and weighted versions of the top cycle and the uncovered set [9].

An interesting related question that goes beyond the computation of possible and necessary winners is the following: when the winners are not yet fully determined, which unknown comparisons need to be learned, or which matches should be played? The construction of a policy tree defining an optimal protocol minimizing the number of questions to be asked or the number of matches to be played, in the worst case or on average, is an even more challenging issue that we leave for further research.

Acknowledgments We thank Gerhard Woeginger for hints towards improving our previous pseudo-polynomial time algorithms for PWS_{BO} and PWS_{MM}. This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR 2312/9-1, BR 2312/10-1, and FI 1664/1-1. Jérôme Lang has been supported by the ANR project ComSoc (ANR-09-BLAN-0305).

References

1. D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. *Information Processing Letters*, 112(5):186–190, 2012.
2. N. Betzler and B. Dorn. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *Journal of Computer and System Sciences*, 76(8): 812–836, 2010.
3. N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proc. of 21st IJCAI*, pages 53–58, 2009.
4. F. Brandt, F. Fischer, and P. Harrenstein. The computational complexity of choice sets. *Mathematical Logic Quarterly*, 55(4):444–459, 2009.
5. M. Brill and F. Fischer. The price of neutrality for the ranked pairs method. In J. Hoffmann and B. Selman, editors, *Proc. of 26th AAAI Conference*. AAAI Press, 2012. Forthcoming.
6. I. Charon and O. Hudry. A survey on the linear ordering problem for weighted or unweighted tournaments. *4OR*, 5(1):5–60, 2007.
7. Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. *Mathematical Social Sciences*, 64(1):74–88, 2012.
8. W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley and Sons, 1998.
9. P. De Donder, M. Le Breton, and M. Truchon. Choosing from a weighted tournament. *Mathematical Social Sciences*, 40(1):85–109, 2000.
10. B. Dutta and J.-F. Laslier. Comparison functions and choice correspondences. *Social Choice and Welfare*, 16(4):513–532, 1999.
11. J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
12. N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. On the evaluation of election outcomes under uncertainty. *Artificial Intelligence*, 189:1–18, 2012.
13. M. Kalech, S. Kraus, G. A. Kaminka, and C. V. Goldman. Practical voting rules with partial information. *Journal of Autonomous Agents and Multi-Agent Systems*, 22(1): 151–182, 2011.
14. W. Kern and D. Paulusma. The computational complexity of the elimination problem in generalized sports competitions. *Discrete Optimization*, 1(2):205–214, 2004.

15. K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. of Multidisciplinary Workshop on Advances in Preference Handling*, pages 124–129. 2005.

16. J. Lang, M. S. Pini, F. Rossi, D. Salvagnin, K. B. Venable, and T. Walsh. Winner determination in voting trees with incomplete preferences and weighted votes. *Journal of Autonomous Agents and Multi-Agent Systems*, 25(1):130–157, 2012.

17. J.-F. Laslier. *Tournament Solutions and Majority Voting*. Springer, 1997.

18. J. E. Peris and B. Subiza. Condorcet choice correspondences for weak tournaments. *Social Choice and Welfare*, 16(2):217–231, 1999.

19. M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Dealing with incomplete agents' preferences and an uncertain agenda in group decision making via sequential majority voting. In *Proc. of 11th KR Conference*, pages 571–578. AAAI Press, 2008.

20. M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Possible and necessary winners in voting trees: Majority graphs vs. profiles. In *Proc. of 10th AAMAS Conference*, pages 311–318, 2011.

21. A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

22. J. H. Smith. Aggregation of preferences with variable electorate. *Econometrica*, 41(6): 1027–1041, 1973.

23. T. N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

24. T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proc. of 22nd AAAI Conference*, pages 3–8. AAAI Press, 2007.

25. L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011.

26. L. Xia, J. Lang, and J. Monnot. Possible winners when new alternatives join: New results coming up! In *Proc. of 10th AAMAS Conference*, pages 829–836, 2011.

Haris Aziz, Markus Brill, Paul Harrenstein, Hans Georg Seedig
Institut für Informatik
Technische Universität München
Garching bei München, Germany
Email: `{aziz,brill,harrenst,seedigh}@in.tum.de`

Felix Fischer
Statistical Laboratory
University of Cambridge
Cambridge, UK
Email: `fischerf@statslab.cam.ac.uk`

Jérôme Lang
LAMSADE
Université Paris-Dauphine
Paris, France
Email: `lang@lamsade.dauphine.fr`

Housing Markets with Indifferences: a Tale of Two Mechanisms

Haris Aziz and Bart de Keijzer

Abstract

The (Shapley-Scarf) housing market is a well-studied and fundamental model of an exchange economy. Each agent owns a single house and the goal is to reallocate the houses to the agents in a mutually beneficial and stable manner. Recently, Alcalde-Unzu and Molis [2011] and Jaramillo and Manjunath [2011] independently examined housing markets in which agents can express indifferences among houses.They proposed two important families of mechanisms, known as TTAS and TCR respectively. We formulate a family of mechanisms which not only includes TTAS and TCR but also satisfies many desirable properties of both families. As a corollary, we show that TCR is strict core selecting (if the strict core is non-empty). Finally, we settle an open question regarding the computational complexity of the TTAS mechanism. Our study also raises a number of interesting research questions.

1 Introduction

Housing markets are fundamental models of exchange economies of goods where the goods could range from dormitories to kidneys [Sönmez and Ünver, 2011]. The classic housing market (also called the Shapley-Scarf Market) consists of a set of agents each of which owns a house and has strict preferences over the set of all houses. The goal is to redistribute the houses to the agents in the most desirable fashion. Shapley and Scarf [1974] showed that a simple yet elegant mechanism called *Gale's Top Trading Cycle (TTC)* is strategy-proof and finds an allocation which is in the core. TTC is based on multi-way exchanges of houses between agents. Since the basic assumption in the model is that agents have strict preferences over houses, TTC is also strict core selecting and therefore Pareto optimal.

Indifferences in preferences are not only a natural relaxation but are also a practical reality in many cases. Many new challenges arise in the presence of indifferences: core stability does not imply Pareto optimality; the strict core can be empty [Quint and Wako, 2004]; and strategic issues need to be re-examined. In spite of these challenges, Alcalde-Unzu and Molis [2011] and Jaramillo and Manjunath [2011] proposed desirable mechanisms for housing markets with indifferences. Alcalde-Unzu and Molis [2011] presented the *Top Trading Absorbing Sets (TTAS)* family of mechanisms which are strategy-proof, core selecting (and therefore individually rational), Pareto optimal, and strict core selecting (if the strict core is non-empty). Independently, Jaramillo and Manjunath [2011] came up with a different family of mechanisms called *Top Cycle Rules (TCR)* which are strategy-proof, core selecting, and Pareto optimal. Whereas it was shown in [Jaramillo and Manjunath, 2011] that each TCR mechanism runs in polynomial time, the time complexity of TTAS was raised as an open problem in [Alcalde-Unzu and Molis, 2011].

We first highlight the commonality of TCR and TTAS by describing a simple class of mechanisms called *Generalized Absorbing Top Trading Cycle (GATTC)* which encapsulates the TTAS and TCR families. It is proved that each GATTC mechanism is core selecting, strict core selecting, and Pareto optimal. As a corollary, TCR is strict core selecting. We note that whereas a GATTC mechanism satisfies a number of desirable properties, the strategy-proofness of a particular GATTC mechanism hinges critically on the order and way of choosing trading cycles. Finally, we settle the computational complexity of TTAS.

By simulating a binary counter, it is shown that a TTAS mechanism can take exponential time to terminate.

2 Preliminaries

Let N be a set of n agents and H a set of n houses. The endowment function $\omega : N \to H$ assigns to each agent the house he originally owns. Each agent has complete and transitive preferences $\succsim_i$ over the houses and $\succsim = (\succsim_1, \ldots, \succsim_n)$ is the preference profile of the agents. The *housing market* is a quadruple $M = (N, H, \omega, \succsim)$. For $S \subseteq N$, we denote $\omega(S) = \{\omega(i) : i \in S\}$ by $\omega(S)$. A function $x : S \to H$ is an *allocation* on $S \subseteq N$ if there exists a bijection π on S such that $x(i) = \omega(\pi(i))$ for each $i \in S$. The goal in housing markets is to re-allocate the houses in a mutually beneficial and efficient way. An allocation is *individually rational (IR)* if $x(i) \succsim_i \omega(i)$. A coalition $S \subseteq N$ *blocks* an allocation x on N if there exists an allocation y on S such that for all $i \in S$, $y(i) \in \omega(S)$ and $y(i) \succ_i x(i)$. An allocation x on N is in the *core (C)* of market M if it admits no blocking coalition. An allocation that is in the core is also said to be *core stable*. An allocation is *weakly Pareto optimal (w-PO)* if N is not a blocking coalition. A coalition $S \subseteq N$ *weakly blocks* an allocation x on N if there exists an allocation y on S such that for all $i \in S$, $y(i) \in \omega(S)$, $y(i) \succsim_i x(i)$, and there exists an $i \in S$ such that $y(i) \succ_i x(i)$. An allocation x on N is in the *strict core (SC)* of market M if it admits no weakly blocking coalition. An allocation that is in the strict core is also said to be *strict core stable*. An allocation is *Pareto optimal (PO)* if N is not a weakly blocking coalition. It is clear that strict core implies core and also Pareto optimality. Core implies weak Pareto optimality and also individual rationality.

A mechanism that always returns a Pareto optimal and (strict) core stable allocation is said to be *Pareto optimal* and *(strict) core-selecting* respectively. A mechanism is *strategy-proof* if for each agent, reporting false preferences to the mechanism will not be beneficial to the agent (i.e., when the agent reports false preferences, he will not end up with a house that he prefers more than the house he would get when he reports his true preferences to the mechanism).

Desirable allocations of housing markets can be computed via a graph-theoretic approach. Each housing market $M = (N, H, \omega, \succsim)$ has a corresponding simple digraph $G(\succsim) = (N \cup H, E)$ such that for each $i \in N$ and $h \in H$, $(i, h) \in E$ if $h \succsim h'$ for all $h' \in H$, and (h, i) if $h = \omega(i)$. In other words, each agent points to his maximally preferred houses and each house points to his owner. An *absorbing set* of a digraph is a strongly connected component from which there are no outgoing edges. Two nodes constitute a *symmetric pair* if there are edges from each node to the other. Both nodes are then called *paired-symmetric*. An absorbing set is *paired-symmetric* if each node belongs to a symmetric pair.

3 GATTC

In this section, we formulate a simple family of mechanisms called *Generalized Absorbing Top Trading Cycle (GATTC)* which is designed for housing markets with indifferences and extends not only TTC but also includes the two families TTAS and TCR. It is based on multi-way exchanges of houses between agents. We will show that GATTC satisfies many desirable properties of housing mechanisms such as being core-selecting and Pareto optimal.

Before we describe GATTC, we will introduce the original TTC mechanism which is for the domain of housing markets with strict preferences. TTC works as follows. For a housing market M with strict preferences, we first construct the corresponding graph $G(\succsim)$ as defined above. Then, we start from an agent and walk arbitrarily along the edges until a cycle is completed. A cycle starting from any agent is of course guaranteed to exist as each node in $G(\succsim)$ has positive outdegree. This cycle is removed from $G(\succsim)$. Within the removed cycle, each agent gets the house he was pointing to. The graph $G(\succsim)$ is *adjusted* so that the remaining agents point to the most preferred houses among the remaining houses. The process is repeated until all the houses and agents are deleted from the graph.[1]

For a housing market with indifferences, TTC can still be used to return a core selecting allocation: break ties arbitrarily and then run TTC [see *e.g.*, Ehlers, 2012]. However such an allocation may not be Pareto optimal [see *e.g.*, Alcalde-Unzu and Molis, 2011, Jaramillo and Manjunath, 2011]. GATTC achieves Pareto optimality and is based on absorbing sets and the concept of a 'good cycle'. A *good cycle* is any cycle in $G(\succsim)$ which contains at least one node that is not paired-symmetric. By *implementing a cycle* we mean reallocating the houses along the cycle. For example consider the cycle $a_0, h_1, a_1, \ldots, h_m, a_m, h_0, a_0$. Then for all $i \in \{0, \ldots, m\}$, house $h_{i+1 \mod m}$ is made to point to a_i. The following is the description of a GATTC mechanism.

GATTC

`Let` $G = G(\succsim)$ `and repeat the following until` G `is empty.`

1. `Repeat the following a finite number of times on` G`:`
   1.1. `Either implement a non-good cycle (if` G `is not empty), or do nothing.`
   1.2. `Either remove a paired-symmetric absorbing set and adjust`[2] G `(if a paired-symmetric absorbing set exists), or do nothing.`
2. `Repeatedly remove paired-symmetric absorbing sets and adjust` G`, until there are no paired-symmetric absorbing sets in` G`.`
3. `If` G `is not empty, implement a good cycle.`

We stress that the choices that a GATTC mechanism makes in steps 1.1. and 1.2. are allowed to be different each time the mechanism executes these steps during the same run. The same holds for the number of times that steps 1.1. and 1.2. are repeated, each time that step 1 is executed. It is not even required that a GATTC mechanism is deterministic: as long as it has the property that the output can always be obtained by a procedure that respects the form above, it is part of the GATTC family.

Example 1 *Consider a housing market $M = (N, H, \omega, \succsim)$ where $N = \{a_1, \ldots, a_5\}$, $H = \{h_1, \ldots, h_5\}$, ω is such that $\omega(a_i) = h_i$ for all $i \in \{1, \ldots, 5\}$, and preferences $\succsim$ are defined as follows:*

agent	a_1	a_2	a_3	a_4	a_5
preferences	h_2	h_3	h_4, h_5	h_1	h_2
	h_1	h_2	h_3	h_5	h_4
				h_4	h_5

Then, if ties are broken in any way, TTC does not return a Pareto optimal allocation. However, GATTC (or TTAS/TCR) returns the following Pareto optimal allocations: $\{\{a_1, h_2\}, \{a_2, h_3\}, \{a_3, h_5\}, \{a_4, h_1\}, \{a_5, h_4\}\}$ or $\{\{a_1, h_1\}, \{a_2, h_3\}, \{a_3, h_4\}, \{a_4, h_5\}, \{a_5, h_2\}\}$. Figure 2 (placed at the end of this paper, due to space constraints) illustrates the first steps in the execution of a GATTC mechanism on this housing market.

Illustration of the first steps of a GATTC mechanism applied to the housing market in Example 1.

[1] Please see Section 2.2 of [Sönmez and Ünver, 2011] for an elegant illustration of how TTC works.

[2] *Adjusting* is defined here in the same way as for the TTC mechanism.

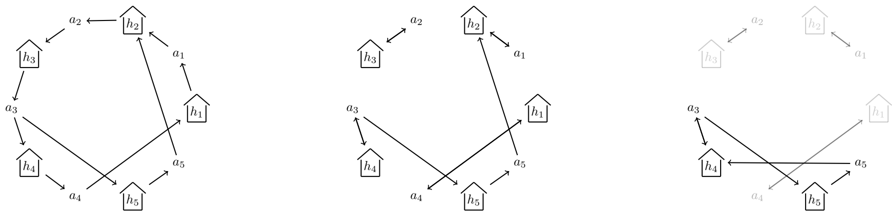


Figure 1: Illustration of the first steps of a GATTC mechanism applied to the housing market in Example 1. The top figure shows the graph as initialized. The algorithm proceeds by executing step 1 zero times, removing no paired-symmetric absorbing sets in step 2 (as there are none), and implementing the cycle $(a_1, h_2, a_2, h_3, a_3, h_4, a_4, h_1, a_1)$ in step 3. The graph after implementing this cycle is shown in the middle figure. Subsequently, the mechanism removes the paired-symmetric absorbing sets, forcing a_5 to point to his second-most preferred houses, i.e., house h_4.

We say that a housing market mechanism is *valid* if it terminates and returns a proper allocation.

Theorem 1 *GATTC is valid, core-selecting, and Pareto optimal.*

Proof: We prove each property separately:

- *Valid:* At the beginning of every step, G has the property that each node has positive out-degree. For non-empty graphs with this property, an absorbing set of cardinality greater than 1 is guaranteed to exist [Kalai and Schmeidler, 1977]. Therefore, if G is not empty, then at step 1.1. there is guaranteed to be a cycle, and at step 3. there is guaranteed to be a good cycle (because there must be an absorbing set that is not paired-symmetric). In each iteration (of steps 1, 2, and 3), if paired-symmetric absorbing sets exist they are removed in Step 2.[3] Also, at least one good cycle is implemented in step 3 which reduces the number of non-paired-symmetric nodes. Therefore, there can be a maximum of $O(n)$ iterations until GATTC terminates. Since each removed house is allocated to the agent it was last pointing to, GATTC returns a proper allocation.
- *Core selecting:* When any agent i is removed from the graph along with his allocated house h, then h is a maximal house for i from among the remaining houses. Therefore i cannot be in a blocking coalition with the agents remaining in the graph.
- *Pareto optimal*: Let S_k be the kth paired-symmetric absorbing set that arises at some point in the GATTC mechanism (and is thus removed from the graph by the GATTC mechanism, and is included accordingly in the allocation produced by the GATTC mechanism). In any allocation x in which none of the players in S_1 are worse off than in the allocation produced by GATTC, the players in S_1 must be allocated to houses in S_1. Taking this as the base case, it follows by easy induction that in x, the players of S_k must be allocated to houses in the kth paired-symmetric absorbing set. Next, suppose that i is a player in S_k for some k. Then no house in S_k is more preferred by i than the house that the GATTC mechanism assigns him to. It follows that no player is strictly better off in x than in the allocation produced by GATTC.

This completes the proof. □

[3] An absorbing set of a graph can be computed in linear time via the algorithm of Tarjan [1972].

Theorem 2 *GATTC is strict core selecting in case the strict core is non-empty.*

Proof: We prove the statement by proving two claims.

Claim 1 *GATTC ensures that if each agent in an absorbing set A can get his maximal house within A, then it will.*

Proof: Define an *inward set* as a set of vertices without edges pointing outward from A. An absorbing set is by definition an inward set. We prove this claim for the more general notion of inward sets. Let A be an inward set that arises at some point in time t during execution of the GATTC mechanism, and assume that each agent can simultaneously get a maximal house in A. If A eventually becomes paired-symmetric, then every agent in A surely gets a maximal house within A. Let us thus assume that A does not eventually become paired-symmetric. Consider the first point in time t' where vertices are removed from A by the mechanism. This point t' exists because the mechanism terminates. All cycles that are implemented in between t and t' either lie completely inside A or completely outside A, because there are no edges pointing from outside A to a vertex in A. It follows that at point t', the removed paired-symmetric absorbing set A' is a strict subset of A. Note that agents in $A \setminus A'$ cannot get a house from within A' without some agent in A' getting a worse house. Hence, by the assumption that each agent in A can get his maximal house within A, it follows that agents in $A \setminus A'$ can still all get a maximal house from within $A \setminus A'$. The proof follows by induction; repeating the same argument on the inward set $A \setminus A'$ that arises when removing A' from the graph. □

Claim 2 *The returned allocation x is in the strict core if and only if for each absorbing set A encountered in the algorithm, each agent in A will get his maximal house in A.*

Proof: ($\Rightarrow$) Assume there is an agent $i \in A$ such that there exists a house h in A for which $h \succsim_i x(i)$. But then i can be involved in a weakly blocking coalition by forming a cycle within A.

($\Leftarrow$) Assume that each agent i in A gets a maximal house from within A. Thus i cannot be part of a blocking coalition. It could still be part of a weakly blocking coalition if an agent i in A had a maximal house h outside A within the remaining graph and there exists a cycle of the form $i, h, \ldots, i$. But this is not possible since A is absorbing. □

From the two claims, the theorem follows. □

We also observe that on the domain of strict preferences, GATTC is equivalent to TTC. The reason is that implementation of any cycle results in a paired-symmetric absorbing set which is then removed from the graph. Ma [1994] proved that for housing markets with strict preferences, a mechanism is strict core selecting if and only if it is individually rational, Pareto optimal, and strategy-proof. On the other hand, we note that in the presence of ties, even if a mechanism is (strict) core selecting, and Pareto optimal, it is not necessarily strategy-proof.

Theorem 3 *Not every GATTC mechanism is strategy-proof.*

Proof Sketch: Consider the following GATTC mechanism in which no non-good cycle is implemented and every good cycle is found in the following way. Consider $a_i \in N$, $h_j \in H$ such that $(a_i, h_j) \in E$, $(h_j, a_i) \notin E$, and a_i and h_j are in a strongly connected component. Then, there exists a shortest path P from h_j to a_i. Find this path P by Dijkstra's shortest path algorithm. Path P gives us a good cycle a_i, h_j, P, a_i.

For this subclass of GATTC, it can be shown that an agent may have incentive to lie about his preferences to obtain a better allocation. Informally, there exist instances of a housing market in which if an agent a does not lie, it may only get a third most preferred house. However, if a points to his second most preferred house h in the graph, it can manage to influence which good cycle is selected and be included in that good cycle. Agent a then gets allocated h. □

4 TTAS and TCR

We now describe the two families of mechanisms in the literature — TTAS [Alcalde-Unzu and Molis, 2011] and TCR [Jaramillo and Manjunath, 2011] — designed for housing markets with indifferences. Both families of mechanisms are extensions of TTC. We will later show that both families are subclasses of GATTC.

TTAS

Fix a priority ranking of the houses; i.e., a complete, transitive and antisymmetric binary relation over H. Construct the graph $G(\succsim)$, and run the following procedure on it (starting with $i = 1$, incrementing i every iteration) until no more agents are remaining in the graph.

Step i

(i.1) Let each remaining agent point to her maximal houses among the remaining ones. Select the absorbing sets of this digraph.

(i.2) Consider the paired-symmetric absorbing sets. Their agents are allocated the house that the agents currently point to in the graph. These absorbing sets are removed from the graph.

(i.3) Consider the remaining absorbing sets. Select for each agent a unique house to point to by using the following criterion: each agent i currently owning house h provisionally points only to the house that i likes most (according to $\succsim_i$) among the houses remaining. Ties are broken by selecting among the candidate houses the one that comes after h in the priority order (if there is no such house, then select among the candidate houses the first house in the priority order).

(i.4) Then, in this subgraph, there is necessarily at least one cycle and no two cycles intersect. Assign (provisionally) to each agent in these cycles the house that he is pointing to, but do not remove them from the graph.

The algorithm terminates when no agents and houses remain, and the outcome is the assignment formed during its execution.

TCR

Consider a priority ranking of the agents; i.e., a complete, transitive and antisymmetric binary relation over A. Do the following until no more agents are left.

1. Departure: A group of agents is chosen to ''depart'' if two conditions are met: i) What each agent in the group holds is among his most preferred houses (among the remaining ones), and ii) All of the most preferred houses (among the remaining ones) of the group are held by them. Once a group departs, each agent in it is assigned what he holds and is removed from the set of remaining agents. In addition, their houses are removed from the remaining houses. There may be another group that can be chosen to depart.

```
   The process continues until there are no more groups that can depart.  If the
   two conditions are not met by any group, then nobody departs.

2. Pointing:  Each agent points to an agent holding one of his top houses (among
   the remaining ones).  Since there may be more than one such agent, the problem
   of figuring whom each agent points to is a complicated one.
   We solve it in stages as follows:

Stage 1 For each remaining j such that j holds the same house that he held in the
        previous step, each i that pointed at j in the previous step points to j
        in the current step.  Of course, this does not apply for the very first
        step.
Stage 2 Each i with a unique top house (among the remaining ones) points to the
        agent holding it.
Stage 3 Each agent who has at least one of his top houses (among the remaining
        ones) held by an unsatisfied agent points to whomever has the highest
        priority among such unsatisfied agents.
Stage 4 Each agent who has at least one of his top houses (among the remaining
        ones) held by a satisfied agent who points to an unsatisfied agent points
        to whomever points to the unsatisfied agent with highest priority.  If
        two or more of his satisfied ''candidates'' point to the unsatisfied
        agent with highest priority, he points to the satisfied candidate with the
        highest priority.
Stage ... And so on.

3. Trading:  Since each remaining agent points to someone, there is at least
   one cycle of remaining agents.  For each such cycle, agents trade according
   to the way that they point and what they hold for the next step is updated
   accordingly.
```

Note that TTAS and TCR mechanisms depend on the priority ordering over H and A respectively. The variation in priority rankings leads to classes of mechanisms rather than a single mechanism. Next, we show that TTAS and TCR are subclasses of GATTC in which cycles are selected via the strict order over houses and agents respectively.

Theorem 4 *GATTC generalizes both the TTAS and TCR families of mechanisms.*

Proof: (*GATTC generalizes TTAS*). (*GATTC generalizes TTAS*). Step i.2 of TTAS corresponds to repeatedly executing step 1.2. (and skipping step 1.1). After that, TTAS may implement a number of non-good cycles. This corresponds in GATTC to executing step 1.1 (skipping step 1.2). However, the proof of Proposition 1 in [Alcalde-Unzu and Molis, 2011] shows that TTAS can never perpetually implement non-good cycles: Either the graph becomes empty, or eventually a good cycle is found and implemented. So executing in TTAS step i.2 to i.4 on iterations where a good cycle is implemented, corresponds to executing steps 3 and 4 of GATTC.

(*GATTC generalizes TCR*). A TCR rule reduces to the GATTC mechanism if zero non-good cycles are implemented in Step 1. and if in Step 3 of GATTC, a good cycle is implemented in the particular way as outlined in the definition as TCR. It is clear from the Step 2 (pointing) of TCR that the way agents are made to point, the cycle induced involves at least one node which is not paired-symmetric. Therefore the cycle in question is a good cycle. □

In contrast to TTAS (which is strict core-selecting), it was not known whether TCR is also strict core-selecting. As a corollary of Theorems 2 and 4, we obtain the following.

Corollary 1 *Each TCR mechanism is strict core selecting (if the strict core is non-empty).*

In the next section, we answer an open question concerning the running time of the TTAS mechanism.

5 Complexity of TTAS

An important property of TTAS is that if an agent i is reallocated a house h during the running of TTAS but i and h are not yet deleted from the graph, then agent i is guaranteed to be finally allocated a house $h' \in H$ such that $h \sim_i h'$ [Lemma 1, Alcalde-Unzu and Molis, 2011]. Therefore the number of symmetric pairs can only increase during the running of the algorithm although it may stay constant in a number of iterations. Alcalde-Unzu and Molis [2011] showed that despite a number of stages in which no obvious progress is being made, TTAS eventually terminates [Proposition 1, Alcalde-Unzu and Molis, 2011]. Although, we know that TTAS terminates and results in a proper allocation, the proof of [Proposition 1, Alcalde-Unzu and Molis, 2011] does not help shed light on how many steps are taken in the running of TTAS.We will show the following.

Theorem 5 *There exists a family of housing markets* $\{M_k = (N_k, H_k, \omega_k, \preceq^k) : k \in \mathbb{N}_{>0}\}$ *with* $|N_k| = |H_k| = 2k+1$*, and corresponding priority rankings* $\{R_k : k \in \mathbb{N}_{>0}\}$ *such that if the TTAS mechanism receives input* M_k *and chooses* R_k *as its priority ranking in step* 0*, then the TTAS mechanism runs for at least* $2^k = 2^{(|N_k|-1)/2}$ *steps until it terminates.*

This theorem shows thus that the TTAS mechanism, according to its current description, does not run in polynomial time. It still might be that for each instance, there is some priority ranking such that the TTAS mechanism runs in polynomial time, but then at least some additional details are needed in the description on how to choose the priority ranking. The algorithm described in Alcalde-Unzu and Molis [2011] is not sufficient.

Proof: The houses and agents of housing market M_k are named as $H_k = \{h_1, h'_1, h_2, h'_2, \ldots, h_k, h'_k, h_{k+1}\}$ and $\{a_1, a'_1, a_2, a'_2, \ldots, a_k, a'_k, a_{k+1}\}$ respectively. In the initial endowment, house h_j is assigned to agent a_j for all $j \in [k+1]$,[4] and house h'_j is assigned to agent a'_j for all $j \in [k]$. The preference profile of agent a_j, $j \in [k]$ is described by two equivalence classes: his class of most preferred houses is $\{h'_j, h_j, h_{j+1}\}$, and the remainder of the houses is in his other equivalence class, i.e., his class of least preferred houses. The preference profile of agent a'_j, $j \in [k]$, is also described by two equivalence classes: His class of most preferred houses is $\{h_j, h'_j, h_1\}$ (so for $j = 1$, this set has cardinality 2), and the remainder of the houses are in the other equivalence class, i.e., his class of least preferred houses. The preference profile of agent a_{k+1} is also described by two equivalence classes: His class of most preferred houses is $\{h_1\}$, and the remainder of the houses is in his other equivalence class, i.e., his class of least preferred houses. The priority ranking R is $(h_1, h'_1, h_2, h'_2, \ldots, h_k, h'_k, h_{k+1})$.

The high level idea of this example is to simulate a binary counter. The graph that the TTAS mechanism maintains will contain a single absorbing set at every step: the entire graph. At every step except the last one, the only agent that prevents the graph from being paired-symmetric will be agent a_{k+1}. We associate bit-strings of length k to the graphs that may arise in some of the steps of the TTAS algorithm: Let $b \in \{0,1\}^k$ be any bit-string of length k, then we define the graph G_b as the graph where for all j,

- a_j and a'_j all point to their set of most preferred houses,

[4] Suppose $x \in \mathbb{N}$, then $[x]$ stands for the set $\{1, \ldots, x\}$.

- if $b_j = 0$, then h_j points to a_j and h'_j points to a'_j.
- if $b_j = 1$, then h_j points to a'_j and h'_j points to a_j.

We prove that for all bit-strings b of length k there is a step i_b such that the graph at the beginning of step i_b is equal to G_b. Because there are 2^k possible bit-strings, it then follows that there are at least 2^k steps before the algorithm terminates.

In order to understand what happens during the execution of the TTAS algorithm on an instance M_j, it will be helpful to look at the example of Figure 2, where the graph at the beginning of every step is shown when we run the TTAS mechanism on M_3.

Let us assume that at the beginning of step i of the execution of the TTAS mechanism, the graph is equal to G_b for some b. We can prove that G_b is strongly connected:

Claim 3 *For each length k bit-string b, G_b is strongly connected.*

Proof: We first show that there is a path from h_1 to every other vertex v.

If $b_1 = 0$, then h_1 points to a_1 and h'_1 points to a'_1. If $b_2 = 0$, then there exists a path $(h_1, a_1, h_2, a_2, h'_2, a'_2)$. If $b_2 = 1$, then there exists a path $(h_1, a_1, h_2, a_2, h'_2, a_2)$.

If $b_1 = 1$, then h_1 points to a'_1 and h'_1 points to a_1. If $b_2 = 0$, then there exists a path $(h_1, a'_1, h'_1, a_1, h_2, a_2, h'_2, a'_2)$. If $b_2 = 1$, then there is a path $(h_1, a'_1, h'_1, a_1, h_2, a'_2, h'_2, a_2)$.

Therefore h_1 has a path to each of the following vertices: $a_1, a_2, h_1, h_2, a'_1, a'_2, h'_1, h'_2$.

Using the same argument, we can see that for each a_j, there is a path to a_{j+1}; for each a'_j, there is a path to a'_{j+1}; for each h_j there is a path to h_{j+1}; for each h'_j, there is a path to h'_{j+1}. Therefore, it holds that: From h_1, there is a path to each a_j for $j \in [k+1]$; From h_1, there is a path to each a'_j for $j \in [k]$; From h_1, there is a path to each h_j for $j \in [k+1]$; and from h_1, there is a path to each h'_j for $j \in [k]$.

Similarly, it can be shown that from every vertex, there is a path to h_1. This completes the argument of the claim.

□

Therefore, G_b has only one absorbing set: the whole of G_b.

Also observe that for all b, G_b is not paired symmetric, because of player $k+1$. From this we conclude that if the graph at the beginning of a step i is equal to G_b, for some $b \in \{0,1\}^k$, then the TTAS mechanism does not terminate at step i, and the mechanism will certainly reach step $i+1$.

For some step i of the TTAS mechanism, and for every agent $a \in N$, let S_a^i denote the set of most preferred houses of a that are ranked lower than the house assigned to a in step i. However, if this set is empty, then define S_a^i to be the set of most preferred houses of a. Let us assume that for step i, the following property holds, which we will call *Property A_i*: for every agent $a \in N$, it holds that the set of most preferred houses of a that have been provisionally assigned to a the least number of times (including 0 times), is S_a^i.

We define a straightforward bijection $c : \{0,1\}^k \to [2^k - 1] \cup \{0\}$ as follows: bit-string b corresponds to the integer $\sum_{j=1}^{k} 2^{j-1} b_j$. We then see that the following happens:

Claim 4 *Let b be a bit-string of length k, suppose that i is a step in the TTAS mechanism such that the graph at step i is equal to G_b, and suppose that Property A_i holds.*

- *If $c(b)$ is even, then the graph at step $i+1$ of the TTAS algorithm is equal to G_{b+1}, and Property A_{i+1} holds.*
- *If $c(b)$ is odd and not equal to $2^k - 1$, then the graph at step $i+2$ of the TTAS algorithm is equal to G_{b+1}, and Property A_{i+2} holds.*

Proof: If $c(b)$ is even, it is easy to see that at the beginning of step $i+1$, the graph will be $G_{c^{-1}(c(b)+1)}$: the only cycle found in part 3 of step i is $(h_1, a_1, h'_1, a'_1, h_1)$. Any other cycles would have to make use of one of the arcs pointing toward h'_1, but that is not possible by the vertex-disjointness property of the cycles in the subgraph used at part 3 of step i. After augmenting G_b according to cycle $(h_1, a_1, h'_1, a'_1, h_1)$, it is easy to check that the graph is equal to G_{b+1}. Also, observe that Property A_{i+1} holds.

If $c(b)$ is odd and not equal to $2^k - 1$, then define j to be the largest index such that $b_{j'} = 1$ for all $j' \leq j$. Then, in part 3 of step i, the cycle $(h_1, a'_1, h'_1, a_1, h_2, a'_2, h'_2, a_2, \ldots, h_j, a'_j, h'_j, a_j, h_{j+1}, a_{j+1}, h'_{j+1}, a'_{j+1}, h_1)$ is found, and no other cycle is found, because otherwise h_1 would be in such a cycle: a contradiction. It is not hard to verify that property A_{i+1} holds, and the graph that now arises at the beginning of step $i+1$ is again a single absorbing set that is not paired symmetric, because of a_{k+1}. Step $i+2$ will therefore certainly be reached, and it can be verified by similar reasoning as before that again a single cycle is found in part 3 of step $i+1$. This cycle is $(h_1, a'_{j+1}, h_{j+1}, a_j, h_j, a_{j-1}, h_{j-1}, a_{j-2}, h_{j-2}, \ldots, a_1, h_1)$. Augmenting the graph on this cycle makes the graph exactly equal to $G_{c^{-1}(c(b)+1)}$. Moreover, Property A_{i+2} holds. □

Property A_1 is certainly satisfied, and the graph at step 1 is $G_{000\ldots}$. By straightforward induction, using the claim above, it follows that for all bit-strings b of length k there is indeed a step i_b such that the graph at the beginning of step i_b is equal to G_b. □

6 Discussion

Properties	TTAS	TCR	GATTC
Core, Pareto optimal	✓	✓	**✔** Th. 1
Strict core (if non-empty)	✓	**✔** Cor. 1	**✔** Th. 2
Strategy-proof	✓	✓	**✗** Th. 3
Polynomial-time	**✗** Th. 5	✓	**✗** Th. 5

Table 1: Housing market mechanisms: new results are in a bolder font.

We analyzed and compared two recently introduced housing market mechanisms. Whereas it was shown that TTAS may take exponential time, TCR was shown to be strict core selecting just like TTAS. The new and old results are summarized in Table 1. Our abstraction from TTAS and TCR to GATTC helps identify the crucial higher level details and commonality of both TTAS and TCR. This leads to simple proofs for properties satisfied by any GATTC mechanism. Whereas core, strict core, and Pareto optimality are properties that can be fulfilled by any GATTC mechanism, additionally satisfying strategy-proofness requires subtlety in choosing which cycles are implemented in which order. This additional complexity leads to an exponential time lower bound in the case of TTAS and a difficulty in having a very simple description in the case of TCR. It is easily seen that GATTC, and in particular TTAS and TCR not only apply to housing markets but also to other extensions such as agents having multiple number of initial endowments or no endowments or there being some social endowments i.e., not owned initially by any agent.

Our study leads to a number of further research questions. It will be interesting to characterize the subset of GATTC mechanisms which are strategy-proof or are both strategy-proof and polynomial-time. Another question is to see whether being a GATTC mechanism is a necessary condition to simultaneously achieve core stability, Pareto optimality and strict core stability. We have seen that all known housing market mechanisms which are core selecting and Pareto optimal are also strict core selecting (if the strict core is non-empty). This raises the question whether every housing market mechanism which is core selecting and Pareto optimal is also strict core selecting (if the strict core is non-empty).

Acknowledgements

This material is based on work supported by the Deutsche Forschungsgemeinschaft under the grant BR 2312/10-1. The authors thank Jorge Alcalde-Unzu, Felix Brandt, Paul Harrenstein, Elena Molis, Hans Georg Seedig, and the anonymous reviewers of AAAI 2012 and COMSOC 2012 for helpful comments. This paper was previously accepted for presentation at AAAI 2012.

References

J. Alcalde-Unzu and E. Molis. Exchange of indivisible goods and indifferences: The top trading absorbing sets mechanisms. *Games and Economic Behavior*, 73(1):1–16, 2011.

L. Ehlers. Top trading with fixed tie-breaking in markets with indivisible goods. Technical report, March 2012.

P. Jaramillo and V. Manjunath. The difference indifference makes in strategy-proof allocation of objects. Technical Report 1809955, SSRN, 2011.

E. Kalai and D. Schmeidler. An admissible set occurring in various bargaining situations. *Journal of Economic Theory*, 14:402–411, 1977.

J. Ma. Strategy-proofness and the strict core in a market with indivisibilities. *International Journal of Game Theory*, 23(1):75–83, 1994.

T. Quint and J. Wako. On houseswapping, the strict core, segmentation, and linear programming. *Mathematics of Operations Research*, 29(4):861–877, 2004.

L. S. Shapley and H. Scarf. On cores and indivisibility. *Journal of Mathematical Economics*, 1(1):23–37, 1974.

T. Sönmez and M. U. Ünver. Matching, allocation, and exchange of discrete resources. In J. Benhabib, M. O. Jackson, and A. Bisin, editors, *Handbook of Social Economics*, volume 1 of *Handbooks in Economics*, chapter 17, pages 781–852. Elsevier, 2011.

R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1 (2):146–160, 1972.

Haris Aziz
Institut für Informatik
Technische Universität München
85748 Garching bei München, Germany
Email: `aziz@in.tum.de`

Bart de Keijzer
Centrum Wiskunde & Informatica (CWI)
1098 XG, Amsterdam, The Netherlands
Email: `B.de.Keijzer@cwi.nl`

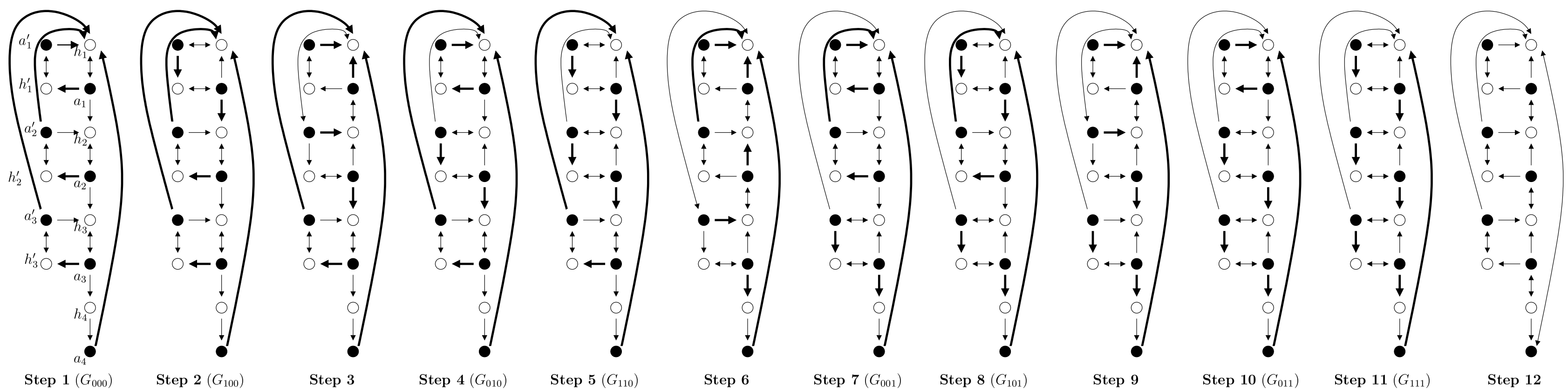


Figure 2: **(Illustrative example for the proof of Theorem 5.)** The graph at the beginning of every step of the TTAS mechanism when it is ran on the instance M_3. Black vertices represent players and white vertices represent houses. When an arc is drawn that has arrows pointing to both its vertices, say vertices a and b, then it stands for the presence of arcs (a, b) and (b, a) in the graph. At the graph for step 1, the names of the vertices are displayed. This is omitted for subsequent steps. In the last step it can be seen that the entire graph is paired symmetric. For every step i except the last one, an arc is displayed in bold in the graph of step i when that arc points from an agent to a house and when that arc is included in the subgraph generated in part 3 of step i (the remaining arcs in this subgraph are all arcs pointing from houses to agents). When in some step, the graph at the beginning of that step equals G_b for some $b \in \{0,1\}^k$, then this is indicated in the figure by the tag "(G_b)" after the step number.

Bribery and Control in Judgment Aggregation[1]

Dorothea Baumeister, Gábor Erdélyi, Olivia J. Erdélyi, and Jörg Rothe

Abstract

In computational social choice, the complexity of changing the outcome of elections via manipulation, bribery, and various control actions, such as adding or deleting candidates or voters, has been studied intensely. Endriss et al. [13, 14] initiated the complexity-theoretic study of problems related to judgment aggregation. We extend their results on manipulation to a whole class of judgment aggregation procedures, and we obtain stronger results by considering not only the classical complexity (NP-hardness) but the parameterized complexity (W[2]-hardness) of these problems with respect to natural parameters. Furthermore, we introduce and study the closely related concepts of bribery and control in judgment aggregation. In particular, we study the complexity of changing the outcome of such procedures via control by adding, deleting, or replacing judges.

1 Introduction

Decision-making processes are often susceptible to various types of interference. In social choice theory and in computational social choice, ways of influencing the outcome of elections—such as manipulation, bribery, and control—have been studied intensely, with a particular focus on the complexity of the related problems (see, e.g., the early work of Bartholdi et al. [2, 1, 3] and the recent surveys and bookchapters by Faliszewski et al. [21, 18], Brandt et al. [5], and Baumeister et al. [4]). In particular, (coalitional) *manipulation* [2, 1, 7] refers to (a group of) strategic voters casting their votes insincerely to reach their desired outcome; in *bribery* [17, 20] an external agent seeks to reach her desired outcome by bribing (without exceeding a given budget) some voters to alter their votes; and in *control* [3, 23, 16] an external agent (usually called the "Chair") seeks to change the structure of an election (e.g., by adding/deleting/partitioning either candidates or voters) in order to reach her desired outcome.

Decision-making mechanisms or systems that are susceptible to strategic behavior, be it from the agents involved as in manipulation or from external authorities or actors as in bribery and control, are obviously not desirable, as that undermines the trust we have in these systems. We therefore have a strong interest in accurately assessing how vulnerable a system for decision-making processes is to these internal or external influences. Unfortunately, in many concrete settings of social choice, "perfect" systems are impossible to exist. For example, the Gibbard–Satterthwaite theorem says that no reasonable voting system can be "strategyproof" [22, 29] (see also the generalization by Duggan and Schwartz [11]), many natural voting systems are not "immune" to most or even all of the standard types of control [3, 23, 16], and Dietrich and List [9] give an analogue of the Gibbard–Satterthwaite theorem in judgment aggregation. To avoid this obstacle, a common approach in computational social choice is to apply methods from theoretical computer science to show that undesirable strategic behavior is blocked, or at least hindered, by the corresponding task being a computationally intractable problem.

Here we focus on judgment aggregation, which is an important framework for collective decision-making. In a judgment aggregation process, we seek to find a collective judgment set from given individual judgment sets over a set of possibly logically interconnected propositions. For further information on judgment aggregation, we refer the reader to the surveys by List and Puppe [26]

[1]This paper extends the results that appeared in *Proceedings of the 2nd International Conference on Algorithmic Decision Theory*, pages 1–15. Springer-Verlag *LNCS #6992*, 2011. This work was supported in part by DFG grant RO 1202/15-1, NRF (Singapore) grant NRF-RF 2009-08, an SFF grant from HHU, and a DAAD PPP/PROCOPE grant.

and by List [25]. This paper follows up the study of manipulation in judgment aggregation initiated by Endriss et al. [14] and it is the first to study bribery and control in judgment aggregation.

In particular, Endriss et al. [13, 14], defined the winner determination problem and the manipulation problem in judgment aggregation and studied their complexity for two important judgment aggregation rules. We extend their complexity-theoretic investigation for manipulation and also introduce various bribery problems in judgment aggregation. Furthermore, we introduce and motivate three types of control in judgment aggregation (namely, control by adding, deleting, or replacing judges), and study their computational complexity. These problems are each closely related to the corresponding problems in voting, yet are specifically tailored to judgment aggregation scenarios.

2 Formal Framework

We follow and extend the judgment aggregation framework described by Endriss et al. [14].

Let *PS* be the set of all propositional variables and $\mathscr{L}_{PS}$ the set of propositional formulas built from *PS*, where the following connections can be used in their usual meaning: disjunction ($\vee$), conjunction ($\wedge$), implication ($\rightarrow$), equivalence ($\leftrightarrow$), and the boolean constants 1 and 0. To avoid double negations, let $\sim\alpha$ denote the complement of α, i.e., $\sim\alpha = \neg\alpha$ if α is not negated, and $\sim\alpha = \beta$ if $\alpha = \neg\beta$. The judges have to judge over all formulas in the *agenda* Φ, which is a finite, nonempty subset of $\mathscr{L}_{PS}$ without doubly negated formulas. The agenda is required to be closed under complementation, i.e., $\sim\alpha \in \Phi$ if $\alpha \in \Phi$. A *judgment set for an agenda* Φ is a subset $J \subseteq \Phi$. It is said to be an *individual judgment set* if it is the set of propositions in the agenda accepted by an individual judge. A *collective judgment set* is the set of propositions in the agenda accepted by all judges as the result of a judgment aggregation procedure. A judgment set J is *complete* if for all $\alpha \in \Phi$, $\alpha \in J$ or $\sim\alpha \in J$; it is *complement-free* if for no $\alpha \in \Phi$, α and $\sim\alpha$ are in J; and it is *consistent* if there is an assignment that makes all formulas in J true. If a judgment set is complete and consistent, it is obviously complement-free. By $\mathscr{J}(\Phi)$ we denote the set of all complete and consistent subsets of Φ.

The famous doctrinal paradox [24] in judgment aggregation shows that if the majority rule is used, the collective judgment set can be inconsistent even if all individual judgment sets are consistent. One way of circumventing the doctrinal paradox is to impose restrictions on the agenda. Endriss et al. [13] studied the question of whether one can guarantee for a specific agenda that the outcome is always complete and consistent. They established necessary and sufficient conditions on the agenda to satisfy these criteria, and they studied the complexity of deciding whether a given agenda satisfies these conditions. They also showed that deciding whether an agenda guarantees a complete and consistent outcome for the majority rule is an intractable problem.

Endriss et al. [14] studied the winner and manipulation problem for two specific judgment aggregation procedures that always guarantee consistent outcomes. In the premise-based procedure, this is achieved by applying the majority rule only to the premises of the agenda, and then to derive the outcome for the conclusions from the outcome of the premises. We will study the complexity of manipulation and control also for the more general class of premise-based quota rules as defined by Dietrich and List [8].

Definition 1 (Premise-based Quota Rule) *The agenda Φ is divided into two disjoint subsets $\Phi = \Phi_p \uplus \Phi_c$, where Φ_p is the set of premises and Φ_c is the set of conclusions. We assume both Φ_p and Φ_c to be closed under complementation. The premises Φ_p are again divided into two disjoint subsets, $\Phi_p = \Phi_1 \uplus \Phi_2$, such that either $\varphi \in \Phi_1$ and $\sim\varphi \in \Phi_2$, or $\sim\varphi \in \Phi_1$ and $\varphi \in \Phi_2$. For each literal $\varphi \in \Phi_1$, define a quota $q_\varphi \in \mathbb{Q}$, $0 \leq q_\varphi < 1$. The quota for the literals $\varphi \in \Phi_2$ is $q'_\varphi = 1 - q_\varphi$.*

A premise-based quota rule is then defined to be a function $PQR : \mathscr{J}(\Phi)^n \rightarrow 2^\Phi$ such that, for

$\Phi = \Phi_p \uplus \Phi_c$, *each profile* $\mathbf{J} = (J_1, \ldots, J_n)$ *is mapped to the judgment set*

$$PQR(\mathbf{J}) = \triangle_q \cup \{\varphi \in \Phi_c \mid \triangle_q \models \varphi\}, \text{ where}$$
$$\triangle_q = \{\varphi \in \Phi_1 \mid \|\{i \mid \varphi \in J_i\}\| > nq_\varphi\} \cup \{\varphi \in \Phi_2 \mid \|\{i \mid \varphi \in J_i\}\| > \lceil nq'_\varphi - 1\rceil\}.$$

To guarantee complete and consistent outcomes for this procedure, it is enough to require that Φ is closed under propositional variables and that Φ_p consists of all literals. The number of affirmations needed to be in the collective judgment set is $\lfloor nq_\varphi + 1\rfloor$ for literals $\varphi \in \Phi_1$ and $\lceil nq'_\varphi\rceil$ for literals $\varphi \in \Phi_2$. Note that $\lfloor nq_\varphi + 1\rfloor + \lceil nq'_\varphi\rceil = n+1$ ensures that either $\varphi \in PQR(\mathbf{J})$ or $\sim\varphi \in PQR(\mathbf{J})$ for every $\varphi \in \Phi$. Note that the quota $q_\varphi = 1$ for a literal $\varphi \in \Phi_1$ is not allowed here, as $n+1$ affirmations were then needed for $\varphi \in \Phi_1$ to be in the collective judgment set, which is impossible. However, $q_\varphi = 0$ is allowed, as in that case $\varphi \in \Phi_1$ needs at least one affirmation and $\sim\varphi \in \Phi_2$ needs n affirmations, which is possible. In the special case of *uniform premise-based quota rules*, there is one quota q for every literal in Φ_1, and the quota $q' = 1-q$ for every literal in Φ_2. We will focus on such rules and denote them by $UPQR_q$. For $q = 1/2$ and the case of an odd number of judges, we obtain the premise-based procedure defined by Endriss et al. [14], and we will denote it by *PBP*.

Furthermore, we will consider yet another variant of premise-based procedure, which was introduced by Dietrich and List [8] and is called *constant premise-based quota rule* and is defined by $CPQR(\mathbf{J}) = \triangle'_q \cup \{\varphi \in \Phi_c \mid \triangle'_q \models \varphi\}$. Here, the number of affirmations needed to be in the set $\triangle'_q$ is a fixed constant. Thus $q_\varphi \in \mathbb{N}$, $0 \le q_\varphi < n$, and $\triangle'_q = \{\varphi \in \Phi_1 \mid \|\{i \mid \varphi \in J_i\}\| > q_\varphi\} \cup \{\varphi \in \Phi_2 \mid \|\{i \mid \varphi \in J_i\}\| > q'_\varphi\}$. Again, to ensure that for every $\varphi \in \Phi$, either $\varphi \in CPQR(\mathbf{J})$ or $\sim\varphi \in CPQR(\mathbf{J})$, we require that $q_\varphi + q'_\varphi = n-1$ for all $\varphi \in \Phi_1$. The uniform variant, $UCPQR_q$, is defined analogously. If the number of judges who take part in the process is fixed, both classes represent the same judgment aggregation procedures. However, we will study control problems where the number of judges can vary. The constant premise-based quota n can then be seen as an upper bound on the highest number of judges possibly participating in the process. This definition is closely related to (a simplified version of) a referendum. Suppose that there is a fixed number of possible participants who are allowed to go to the polls, and there is a fixed number of affirmations needed for a certain decision, independent of the number of people who are actually participating. Of course, this number may depend on the number of possible participants, for example 20% of them.

3 Motivation for Control in Judgment Aggregation

We study three types of control for judgment aggregation. So far control has been studied extensively for voting systems (see, e.g., [3, 23, 4, 16]), where control is normally perceived as dishonest and thus as an undesired behavior. Therefore, this research focuses on finding ways to avoid it. Looking at real-world examples, this point of view is not always justified; in fact, some "control" attempts may be justified by fairly decent considerations (e.g., excluding children from elections is some reasonable kind of exerting control). Nevertheless, one is well advised to be aware of control attempts, since their objective is indeed frequently enough abusive (e.g., excluding voters from elections based on racial or gender grounds, as is still common in certain countries, is abusive and unacceptable). If control is generally possible, one way of circumventing it is to study the computational complexity of the underlying decision problems. If it turns out to be NP-hard, the desired control action can, in general, not be performed in polynomial time, unless P = NP. For practical purposes, showing hardness in appropriate typical-case models is even more useful, but also more challenging [28]. As motivation for studying control in judgment aggregation, we will now illustrate the three different control types for judgment aggregation considered in this paper with some examples from the American jury trial system and international arbitration.

Adding Judges: This first control type is analogous to control by adding voters in elections. An example for this control setting can be found in the field of international arbitration, which is becoming increasingly important as an alternative dispute resolution method to litigations conducted by national courts. Parties of arbitration proceedings may choose to entrust a single arbitrator with deciding their dispute. They might, however, also opt for the appointment of several arbitrators and thereby control the arbitral decision-making process by adding judges.[2] Mostly they do so because they feel that due to the complicated nature of the matter or for some other reason, a tribunal with several arbitrators is better suited to arbitrate their case. Their action may also be motivated by the hope of being able to appoint an arbitrator sympathetic to their arguments.

Deleting judges: Also very natural is the problem of control by deleting judges as it is a commonly applied method in both jury trials and international arbitration. The empaneling procedure of a jury for a trial is basically a control process via deleting judges and works roughly as follows. First, a certain number of potential jurors is summoned at the place of trial. In the next stage of the selection procedure, all or part of them are subjected to the so-called "voir dire" process, i.e., a questioning by the trial judge and/or the attorneys aiming to obtain information about their person. Admittedly, the purpose of collecting this information is to determine whether they can be impartial, which is a well-justified purpose; but again, attorneys may use it for another reason, namely to indoctrinate prospective jurors laying a foundation for arguments they later intend to make. Driven by good or bad intentions, the lawyers may then challenge jurors for cause, that is, by arguing that and for what reason the juror in question is impartial. The trial judge decides over the attorneys' challenges for cause, moreover she may excuse further jurors due to social hardship. Finally, the lawyers may challenge a limited number of potential jurors peremptorily, i.e., without having to justify their reason for doing so. Peremptory challenges are legitimate and useful means of eliminating such jurors that are either presumably biased but the bias cannot be proved to the extent necessary for challenging them for cause, or are for some other reason undesirable. Because their use does not require any explanation, such challenges can also be easily abused; especially until the introduction of the Batson rule, peremptory challenges were often exercised in discriminatory ways, mostly on racial grounds, violating the equal protection rights of jurors. As we can see, deleting judges/jurors is a central part of the empaneling procedure. However, since the total number of jurors is fixed, a new juror needs to be appointed for each deleted juror, which motivates the next scenario.

Replacing judges: Control by replacing judges is used in international arbitration when the parties successfully challenge an arbitrator leading to her disqualification and the subsequent appointment of a substitute arbitrator. The institution of challenge is designed to serve as a tool for parties of arbitral proceedings to remove arbitrators posing a possible threat to the integrity of the proceedings. It may be based on several grounds; arbitrators are most commonly challenged because of doubts regarding their impartiality or independence.[3] Challenges are, however, occasionally used as "black art" or "guerrilla tactics" with a view to achieve dishonest purposes, such as eliminating arbitrators that are likely to render an unfavorable award or to delay the proceedings to evade, or at least postpone, an anticipated defeat.

Control by replacing judges can be seen as a combined action of control by deleting judges and control by adding judges. For a related general model in voting theory, we refer to the work of Faliszewski et al. [19] on "multimode control attacks."

4 Problem Definitions

Bribery problems in voting theory, as introduced by Faliszewski et al. [17] (see also, e.g., [12, 20]), model scenarios in which an external actor seeks to bribe some of the voters to change their votes such that a distinguished candidate becomes the winner of the election. In judgment aggregation it is not the case that one single candidate wins, but there is a decision for every formula in the agenda.

[2] See, for instance, Articles 37–40 of the ICSID Convention and Rules 1–4 of the ICSID Arbitration Rules, Articles 11–12 of the ICC Arbitration Rules, or Articles 7–10 of the UNCITRAL Arbitration Rules.

[3] For rules regarding the challenge, disqualification, and replacement of arbitrators, see Articles 56–58 of the ICSID Convention, Rules 9–11 of the ICSID Arbitration Rules, Articles 14–15 of the ICC Arbitration Rules, and Articles 12–14 of the UNCITRAL Arbitration Rules.

So the external actor might seek to obtain exactly his or her desired collective outcome by bribing the judges, or he or she might be interested only in the desired outcome of some formulas in Φ. The exact bribery problem is then defined as follows for a given aggregation procedure F.

F-EXACT BRIBERY	
Given:	An agenda Φ, a profile $\mathbf{T} \in \mathscr{J}(\Phi)^n$, a consistent and complement-free judgment set J (not necessarily complete) desired by the briber, and a positive integer k.
Question:	Is it possible to change up to k individual judgment sets in $\mathbf{T}$ such that for the resulting new profile $\mathbf{T}'$ it holds that $J \subseteq F(\mathbf{T}')$?

Note that if J is a complete judgment set then the question is whether $J = F(\mathbf{T}')$.

Since in the case of judgment aggregation there is no winner, we also adopt the approach Endriss et al. [14] used to define the manipulation problem in judgment aggregation. In their definition, an outcome (i.e., a collective judgment set) is more desirable for the manipulator if its Hamming distance to the manipulator's desired judgment set is smaller, where for an agenda Φ the Hamming distance $H(J,J')$ between two complete and consistent judgment sets $J,J' \in \mathscr{J}(\Phi)$ is defined as the number of positive formulas in Φ on which J and J' differ. The formal definition of the manipulation problem in judgment aggregation is as follows, for a given aggregation procedure F.

F-MANIPULATION	
Given:	An agenda Φ, a profile $\mathbf{T} \in \mathscr{J}(\Phi)^{n-1}$, and a consistent and complete judgment set J desired by the manipulator.
Question:	Does there exist a judgment set $J' \in \mathscr{J}(\Phi)$ such that $H(J,F(\mathbf{T},J')) < H(J,F(\mathbf{T},J))$?

A specific judgment aggregation procedure is called *strategyproof* if a manipulator can never benefit from reporting an insincere preference. Now, we can give the formal definition of bribery in judgment aggregation, where the briber seeks to obtain a collective judgment set having a smaller Hamming distance to the desired judgment set, then the original outcome has. In bribery scenarios, we extend the above approach of Endriss et al. [14] by allowing that the desired outcome for the briber may be an incomplete (albeit consistent and complement-free) judgment set. This reflects a scenario where the briber may be interested only in some part of the agenda. The definition of Hamming distance is extended accordingly as follows. Let Φ be an agenda, $J \in \mathscr{J}(\Phi)$ be a complete and consistent judgment set, and $J' \subseteq \Phi$ be a consistent and complement-free judgment set. The *Hamming distance $H(J,J')$ between J and J'* is defined as the number of formulas from J' on which J does not agree: $H(J,J') = \|\{\varphi \mid \varphi \in J' \wedge \varphi \notin J\}\|$. Observe that if J' is also complete, this extended notion of Hamming distance coincides with the notion Endriss et al. [14] use.

F-BRIBERY	
Given:	An agenda Φ, a profile $\mathbf{T} \in \mathscr{J}(\Phi)^n$, a consistent and complement-free judgment set J (not necessarily complete) desired by the briber, and a positive integer k.
Question:	Is it possible to change up to k individual judgment sets in $\mathbf{T}$ such that for the resulting new profile $\mathbf{T}'$ it holds that $H(F(\mathbf{T}'),J) < H(F(\mathbf{T}),J)$?

Faliszewski et al. [20] introduced microbribery for voting systems. We adopt their notion so as to apply to judgment aggregation. In microbribery for judgment aggregation, if the briber's budget is k, he or she is not allowed to change up to k entire judgment sets but instead can change up to k premise entries in the given profile (the conclusions change automatically if necessary). We will denote this problem by F-MICROBRIBERY, and the exact variant by F-EXACT MICROBRIBERY.

We will now formally define the underlying decision problems for the complexity-theoretic study of control in judgment aggregation, closely related to the corresponding problems in elections. For a given judgment aggregation procedure F, the problem of control by adding judges is defined as follows:

F-CONTROL BY ADDING JUDGES	
Given:	An agenda Φ, complete profiles $\mathbf{T} \in \mathscr{J}(\Phi)^n$ and $\mathbf{S} \in \mathscr{J}(\Phi)^{\|\mathbf{S}\|}$, a positive integer k, and a consistent and complement-free judgment set J (not necessarily complete).
Question:	Is there a subset $\mathbf{S}' \subset \mathbf{S}$, $\|\mathbf{S}'\| \leq k$, such that $H(J, F(\mathbf{T} \cup \mathbf{S}')) < H(J, F(\mathbf{T}))$?

If we consider the variant F-EXACT CONTROL BY ADDING JUDGES, we ask if there is a subset $\mathbf{S}' \subset \mathbf{S}$, $\|\mathbf{S}'\| \leq k$, such that $J \subseteq F(\mathbf{T} \cup \mathbf{S}')$.

Control by deleting judges is defined as follows for a given judgment aggregation procedure F:

F-CONTROL BY DELETING JUDGES	
Given:	An agenda Φ, a complete profile $\mathbf{T} \in \mathscr{J}(\Phi)^n$, a positive integer k, and a consistent and complement-free judgment set J (not necessarily complete).
Question:	Is there a subset $\mathbf{T}' \subset \mathbf{T}$ with $\|\mathbf{T}'\| \leq k$ such that $H(J, F(\mathbf{T} \setminus \mathbf{T}')) < H(J, F(\mathbf{T}))$?

The exact variant is defined analogously to the case of adding judges.

The new control problem we introduce here is specific to judgment aggregation. It considers the case where some judges may be replaced (see our motivating examples in Section 3):

F-CONTROL BY REPLACING JUDGES	
Given:	An agenda Φ, complete profiles $\mathbf{T} \in \mathscr{J}(\Phi)^n$ and $\mathbf{S} \in \mathscr{J}(\Phi)^{\|\mathbf{S}\|}$, a positive integer k, and a consistent and complement-free judgment set J (not necessarily complete).
Question:	Are there subsets $\mathbf{T}' \subset \mathbf{T}$ and $\mathbf{S}' \subset \mathbf{S}$, with $\|\mathbf{T}'\| = \|\mathbf{S}'\| \leq k$ such that $$H(J, F((\mathbf{T} \setminus \mathbf{T}') \cup \mathbf{S}')) < H(J, F(\mathbf{T}))?$$

Define F-EXACT CONTROL BY REPLACING JUDGES analogously to the exact variants of the adding and deleting judges problems. To study the computational complexity of adding, deleting, and replacing judges, we adopt the terminology introduced in [3] for control problems in voting and adapt it to judgment aggregation. Let F be an aggregation procedure and let $\mathscr{C}$ be a given control type. F is said to be *immune* to control by $\mathscr{C}$ if it is never possible for an external person to successfully control the judgment aggregation procedure via $\mathscr{C}$-control. F is said to be *susceptible* to control by $\mathscr{C}$ if it is not immune. F is said to be *resistant* to control by $\mathscr{C}$ if it is susceptible and the corresponding decision problem is NP-hard. F is said to be *vulnerable* to control by $\mathscr{C}$ if it is susceptible and the corresponding decision problem is in P.

We assume that the reader is familiar with the basic concepts of complexity theory and with complexity classes such as P and NP; see, e.g., [27]. Downey and Fellows [10] introduced *parameterized* complexity theory; in their framework it is possible to do a more fine-grained multi-dimensional complexity analysis. In particular, NP-complete problems may be easy (i.e., fixed-parameter tractable) with respect to certain parameters confining the seemingly unavoidable combinatorial explosion. If this parameter is reasonably small, a fixed-parameter tractable problem can be solved efficiently in practice, despite its NP-hardness. Formally, a *parameterized decision problem* is a set $L \subseteq \Sigma^* \times N$, and we say it is *fixed-parameter tractable* (FPT) if there is a constant c such that for each input (x,k) of size $n = |(x,k)|$ we can determine in time $O(f(k) \cdot n^c)$ whether (x,k) is in L, where f is a function depending only on the parameter k. The main hierarchy of parameterized complexity classes is: $\mathrm{FPT} = \mathrm{W}[0] \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots \subseteq \mathrm{W}[\ell] \subseteq \mathrm{XP}$.

In our results, we will focus on only the class W[2], which refers to problems that are considered to be fixed-parameter intractable. In order to show that a parameterized problem is W[2]-hard, we will give a parameterized reduction from the W[2]-complete problem k-DOMINATING SET (see [10]). We say that a parameterized problem A *parameterized reduces* to a parameterized problem B if each instance (x,k) of A can be transformed in time $\mathscr{O}(g(k) \cdot |x|^c)$ (for some function g and some constant c) into an instance (x',k') of B such that $(x,k) \in A$ if and only if $(x',k') \in B$, where $k' = g(k)$. Note that $g(k) \equiv c$ may also be a constant function not depending on k.

In our proofs we will make use of three different problems. First, we will use the NP-complete problem EXACT COVER BY 3-SETS (X3C for short), where an instance consists of a given set $X = \{x_1, \ldots, x_{3m}\}$ and a collection $C = \{C_1, \ldots, C_n\}$ of 3-element subsets of X, and the question is whether there is an *exact cover for* X, i.e., a subcollection $C' \subseteq C$ such that every element of X occurs in exactly one member of C'. We will also use the DOMINATING SET problem, where we are given a graph $G = (V,E)$ and a positive integer k, and the question is whether there is a *dominating set for G of size at most k*, i.e., whether there is a subset $V' \subseteq V$, $\|V\| \leq k$, such that for each $v \in V$, either $v \in V'$ or there is a $w \in V'$ with $\{v,w\} \in E$. DOMINATING SET is NP-complete and, when parameterized by the upper bound k on the size of the dominating set, its parameterized variant (denoted by k-DOMINATING SET, to be explicit) is W[2]-complete [10]. Finally, we will also use the following problem for our parameterized complexity results:

OPTIMAL LOBBYING	
Given:	An $m \times n$ 0-1 matrix L (whose rows represent the voters, whose columns represent the referenda, and whose 0-1 entries represent No/Yes votes), a positive integer $k \leq m$, and a target vector $x \in \{0,1\}^n$.
Question:	Is there a choice of k rows in L such that by changing the entries of these rows the resulting matrix has the property that, for each j, $1 \leq j \leq n$, the jth column has a strict majority of ones (respectively, zeros) if and only if the jth entry of the target vector x of The Lobby is one (respectively, zero)?

OPTIMAL LOBBYING has been introduced and, parameterized by the number k of rows The Lobby can change, shown to be W[2]-complete by Christian et al. [6] (see also [15] for a more general framework and more W[2]-hardness results).

Note that a multiple referendum as in OPTIMAL LOBBYING can be seen as the special case of a judgment aggregation scenario where the agenda is closed under complementation and propositional variables and contains only premises and where the majority rule is used for aggregation. For illustration, consider the following simple example of a multiple referendum. Suppose the citizens of a town are asked to decide by a referendum whether two projects, A and B (e.g., a new hospital and a new bridge), are to be realized. Suppose the building contractor (who, of course, is interested in being awarded a contract for both projects) sets some money aside to attempt to influence the outcome of the referenda, by bribing some of the citizens without exceeding this budget. Observe that an *PBP*-EXACT BRIBERY instance with only premises in the agenda and with a complete desired judgment set J is nothing other than an OPTIMAL LOBBYING instance, where J corresponds to The Lobby's target vector.[4] Requiring the citizens to give their opinion only for the premises A and B of the referendum and not for the conclusion (whether both projects are to be realized) again avoids the doctrinal paradox. Again, the citizens might also vote strategically in these referenda. Both projects will cost money, and if both projects are realized, the amount available for each must be reduced. Some citizens may wish to support some project, say A, and may be unhappy with reducing the amount for A due to both projects being realized. They might even prefer none of the projects being realized over only B being realized. For them it is natural to consider the possibility of reporting insincere votes (provided they know how the others will vote); this may turn out to be more advantageous for them, as then they can possibly prevent that both projects are realized.

[4] Although exact bribery in judgment aggregation generalizes optimal lobbying in the sense of Christian et al. [6] (which is different from bribery in voting, as defined by Faliszewski et al. [17]), we will use the term "bribery" rather than "lobbying" in the context of judgment aggregation.

5 Results

We start by extending the result of Endriss et al. [14] that *PBP*-MANIPULATION is NP-complete. We study a parameterized version of the manipulation problem and establish a W[2]-hardness result with respect to the uniform premise-based quota rule. Due to space restrictions all proofs except one will be omitted.

Theorem 2 *For each rational quota q,* $0 \leq q < 1$ *and for any fixed number* $n \geq 3$ *of judges,* $UPQR_q$-MANIPULATION *is* W[2]*-hard when parameterized by the maximum number of changes in the premises needed in the manipulator's judgment set.*

Since the reduction is from the NP-complete problem DOMINATING SET, NP-completeness of $UPQR_q$-MANIPULATION, $0 \leq q < 1$, for any fixed number $n \geq 3$ of judges follows immediately from the proof of Theorem 2. Note that NP-hardness of $UPQR_q$-MANIPULATION could have also been shown by a modification of the proof of Theorem 2 in [14], but this reduction would not be appropriate to establish W[2]-hardness, since the corresponding parameterized version of SAT is not known to be W[2]-hard.

As mentioned above, studying the case of a fixed total number of judges is very natural. The second parameter we have considered for the manipulation problem in Theorem 2 is the "maximum number of changes in the premises needed in the manipulator's judgment set." Hence this theorem shows that the problem remains hard even if the number of premises the manipulator can change is bounded by a fixed constant. This is also very natural, since the manipulator may wish to report a judgment set that is as close as possible to his or her sincere judgment set, because for a completely different judgment set it might be discovered too easily that he was judging strategically.

In contrast to the hardness results stated in Theorem 2, the following proposition shows that, depending on the agenda, there are cases in which manipulation for $UPQR_q$, $0 \leq q < 1$, is outright impossible, and thus $UPQR_q$-MANIPULATION is trivially in P.

Proposition 3 *If the agenda contains only premises then* $UPQR_q$, $0 \leq q < 1$, *is strategyproof.*

NP-completeness for $UPQR_q$-MANIPULATION with a fixed number of judges, which is stated in Theorem 2, implies that there is little hope to find a polynomial-time algorithm for the general problem even when the number of participating judges is fixed. However, Proposition 3 tells us that if the agenda is simple and contains no conclusions, $UPQR_q$ is even strategyproof.

Now we will study the complexity of various bribery problems for the premise-based procedure *PBP*, i.e., $UPQR_{1/2}$ for an odd number of judges. We will establish NP-completeness for bribery, microbribery, and exact microbribery, and a W[2]-hardness result for exact bribery with respect to a natural parameter. We start with bribery.

Theorem 4 *PBP*-BRIBERY *is* NP*-complete, even when the total number of judges (*$n \geq 3$ *odd) or the number of judges that can be bribed is a fixed constant.*

Next, we turn to microbribery. Here the briber can change only up to a fixed number of entries in the individual judgment sets. We again prove NP-completeness when the number of judges or the number of microbribes allowed is a fixed constant.

Theorem 5 *PBP*-MICROBRIBERY *is* NP*-complete, even when the total number of judges (*$n \geq 3$ *odd) or the number of microbribes allowed is a fixed constant.*

Theorem 6 *PBP*-EXACT BRIBERY *is* W[2]*-hard when parameterized by the number of judges that can be bribed.*

This result follows from the fact that OPTIMAL LOBBYING is a special case of *PBP*-EXACT BRIBERY. Note that W[2]-hardness with respect to any parameter directly implies NP-hardness for the corresponding unparameterized problem, so *PBP*-EXACT BRIBERY is also NP-complete; all (unparameterized) problems considered here are easily seen to be in NP.

Theorem 7 *PBP*-EXACT MICROBRIBERY *is* NP-*complete, even when the total number of judges ($n \geq 3$ odd) or the number of microbribes allowed is a fixed constant.*

As for the manipulation problem, Theorems 4, 5, and 7 are concerned with a fixed number of judges. It turns out that even in this case BRIBERY, MICROBRIBERY, and EXACT MICROBRIBERY are NP-complete for *PBP*. Furthermore, we consider the case of a fixed number of judges allowed to bribe for *PBP*-BRIBERY, the corresponding parameter for its exact variant, and the case where the number of microbribes allowed is a fixed constant for *PBP*-MICROBRIBERY and its exact variant. Both parameters concern the budget of the briber. Since the briber aims at spending as little money as possible, it is also natural to consider these cases. But again, NP-completeness was shown even when the budget is a fixed constant and in one case W[2]-hardness for this parameter, so bounding the budget does not help to solve the problem easily. Although the exact microbribery problem is computationally hard in general for the aggregation procedure *PBP*, there are some interesting naturally restricted instances where it is computationally easy.

Theorem 8 *If the desired judgment set J is complete or if the desired judgment set is incomplete but contains all of the premises or only premises, then PBP*-EXACT MICROBRIBERY *is in* P*.*

In the last part of this section we study control in judgment aggregation. In the manipulation and bribery problems studied in this paper the number of participating judges is constant and hence uniform premise-based quota rules and uniform constant premise-based quota rules describe the same judgment aggregation procedures. However, this is not the case if the number of participating judges is *not* fixed, as in control by adding or deleting judges. For the uniform premise-based quota rule the number of affirmations needed to be in the collective judgment set varies with the number of judges, whereas for the constant premise-based quota rule the number of affirmations remains the same regardless of the number of judges participating. Since the number of participating judges varies for both control by adding and by deleting judges, we study these problems with respect to both judgment aggregation procedures.

We will first consider the uniform constant premise-based quota rule and show NP-hardness of $UCPQR_q$ for control by adding and by deleting judges in the Hamming distance based and in the exact variant.

Theorem 9 *For each admissible value of q, $UCPQR_q$ is resistant to* CONTROL BY ADDING JUDGES *and to* EXACT CONTROL BY ADDING JUDGES*.*

Theorem 10 *For each admissible value of q, $UCPQR_q$ is resistant to* CONTROL BY DELETING JUDGES *and to* EXACT CONTROL BY DELETING JUDGES*.*

Now we turn to the results for the uniform premise-based quota rule in the case of control by adding and by deleting judges. Here we only consider $UPQR_{1/2}$, which equals the premise-based procedure *PBP* defined by Endriss et al. [14] for an odd number of judges. We show NP-hardness for control by adding and by deleting judges in both problem variants.

Theorem 11 *$UPQR_{1/2}$ is resistant to* EXACT CONTROL BY ADDING JUDGES *and to* CONTROL BY ADDING JUDGES*.*

Proof. Membership in NP is obvious for both problems. Again, we show NP-hardness for $UPQR_{1/2}$-EXACT CONTROL BY ADDING JUDGES only and $UPQR_{1/2}$-CONTROL BY ADDING

JUDGES at the same time, by a reduction from the NP-complete problem X3C. Given an X3C instance (X,C) with $X = \{x_1,\dots,x_{3m}\}$ and $C = \{C_1,\dots,C_n\}$, define the following judgment aggregation scenario. The agenda Φ contains $\{\alpha_0,\alpha_1,\dots,\alpha_{3m}\}$ and their negations. The quota is $1/2$ for every positive literal. The profile of the individual judgment sets initially taking part in the process is $\mathbf{T} = (T_1,\dots,T_{m+1})$ with $T_1 = \{\alpha_0,\alpha_1,\dots,\alpha_{3m}\}$, $T_i = \{\neg\alpha_0,\alpha_1,\dots,\alpha_{3m}\}$, $2 \leq i \leq m$, and $T_{m+1} = \{\neg\alpha_0,\neg\alpha_1,\dots,\neg\alpha_{3m}\}$. The profile of the judges who can be added is $\mathbf{S} = (S_1,\dots,S_n)$ with $S_i = \{\alpha_0,\alpha_j,\neg\alpha_\ell \mid x_j \in C_i, x_\ell \notin C_i, 1 \leq j,\ell \leq 3m\}$. The maximum number of judges from $\mathbf{S}$ who can be added is m. The desired outcome of the external person is $J = \{\alpha_0,\alpha_1,\dots,\alpha_{3m}\}$. Then it holds, that there is a profile $\mathbf{S}' \subseteq \mathbf{S}$, $\|\mathbf{S}'\| \leq m$, such that $H(J,F(\mathbf{T}\cup\mathbf{S}')) < H(J,F(\mathbf{T}))$ if and only if there is an exact cover for the given X3C instance. The collective judgment set for $UPQR_{1/2}(\mathbf{T})$ is $\{\neg\alpha_0,\alpha_1,\dots,\alpha_{3m}\}$. Observe that $H(J,F(\mathbf{T})) = 1$, since the only difference lies in α_0. Hence, $F(\mathbf{T}\cup\mathbf{S}')$ must be exactly J, and the reduction will hold for both problems at hand.

($\Leftarrow$) Assume that there is an exact cover $C' \subseteq C$ for the given X3C instance (X,C). Then the profile $\mathbf{S}'$ contains those judges S_i with $C_i \in C'$. The total number of judges is then $2m+1$. The number of affirmations needed to be in the collective judgment set is strictly greater than $m + (1/2)$, so $m+1$ affirmations are needed. Note that α_0 gets one affirmation from the judges in $\mathbf{T}$ and m affirmations from the judges in $\mathbf{S}'$. Every α_i, $1 \leq i \leq 3m$, gets m affirmations from the judges in $\mathbf{T}$ and one affirmation from a judge in $\mathbf{S}'$. Hence, the collective judgment set is J, as desired.

($\Rightarrow$) Assume that there is a profile $\mathbf{S}'$ with $\|\mathbf{S}'\| \leq m$ such that $UPQR_{1/2}(\mathbf{T}\cup\mathbf{S}') = J$. Since α_0 is contained in the collective judgment set it must receive enough affirmations of the judges in $\mathbf{S}'$. Adding less than m new affirmations for α_0 is not enough, since $m-1 \leq (2m)(1/2)$, but since $(2m+1)(1/2) < m+1$, m new affirmations are enough. As above, if there is a total number of $2m+1$ judges then the number of affirmations needed for a positive formula to be in the collective judgment set is $m+1$. Since the α_i, $1 \leq i \leq 3m$, receive only m affirmations from $\mathbf{T}$, they must all get one additional affirmation from $\mathbf{S}'$. Since $\|\mathbf{S}'\| \leq m$ and every judge affirms of exactly four formulas, including α_0, the sets C_i corresponding to the judges in $\mathbf{S}'$ must form an exact cover for the given X3C instance. ❑

One important point regarding the proof of Theorem 11 is that the agenda contains only premises. For $UPQR_{1/2}$-EXACT CONTROL BY DELETING JUDGES, the proof of Theorem 12 below also establishes NP-hardness even if the agenda contains only premises. By contrast, in Proposition 3 we showed that if the agenda contains only premises then $UPQR_q$ is strategyproof (thus, $UPQR_q$-MANIPULATION is in P) for each rational quota q, $0 \leq q < 1$, and in Theorem 5 we showed that $UPQR_{1/2}$-EXACT MICROBRIBERY is also in P if the desired judgment set contains only premises.

Theorem 12 *$UPQR_{1/2}$ is resistant to* EXACT CONTROL BY DELETING JUDGES *and* CONTROL BY DELETING JUDGES.

In contrast to $UPQR_{1/2}$-CONTROL BY ADDING JUDGES it remains open whether $UPQR_{1/2}$-CONTROL BY DELETING JUDGES is still NP-complete if the agenda contains only premises.

Unlike for manipulation and bribery, we have not been able to identify natural restrictions for which one of our NP-hard control problems can be solved in polynomial time.

Finally, we consider CONTROL BY REPLACING JUDGES. In contrast to the problems of control by adding and by deleting judges, the number of judges here is constant, just as in the corresponding manipulation and bribery problems for judgment aggregation. Thus, there is no difference between the uniform constant premise-based quota rule and the uniform premise-based quota rule. The following theorem implies NP-completeness for both classes of rules.

Theorem 13 *For each rational quota q, $0 \leq q < 1$, $UPQR_q$ is resistant to* EXACT CONTROL BY REPLACING JUDGES *and* CONTROL BY REPLACING JUDGES.

To conclude, we mention some possible future research questions. First, we have introduced some very natural control problems for judgment aggregation. Are there any others? Second, it would be very interesting to complement our NP-hardness results by typical-case analyses, as has been done for voting problems (see the survey [28]). Third, from all W[2]-hardness results we immediately obtain the corresponding NP-hardness results, and since all problems considered are easily seen to be in NP, we have NP-completeness results. It remains open, however, whether one can also obtain matching upper bounds in terms of parameterized complexity. We suspect that all W[2]-hardness results in this paper in fact can be strengthened to W[2]-completeness results. Finally, note that we have considered only "constructive" control scenarios. For voting problems, constructive control means that the Chair's goal is to make some candidate win, whereas "destructive" control [23] refers to making any other than the most hated candidate win the election. Constructive control in judgment aggregation, however, means that we seek an outcome *closer to the desired outcome*, or *exactly the desired outcome*. Note that defining destructive variants of control by adding, deleting, or replacing judges would thus lead to the same definitions as for their constructive counterparts: We have an undesired (possibly partial) judgment set $J \in \mathscr{J}(\Phi)$ and seek an outcome with a smaller Hamming distance to the complement of J than from the original outcome to the complement of J, but replacing the (partial) judgment set J with its complement leads to essentially the same question, as the complement of a partial judgment set J is simply the negation of the formulas in J. Therefore, it does not make sense to distinguish between constructive and desctructive control.

Acknowledgments: We thank the reviewers for helpful comments.

References

[1] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[2] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[3] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8/9):27–40, 1992.

[4] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. In J. Laslier and R. Sanver, editors, *Handbook on Approval Voting*, chapter 10, pages 199–251. Springer, 2010.

[5] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiß, editor, *Multiagent Systems*. MIT Press, second edition, 2012. To appear.

[6] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.

[7] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[8] F. Dietrich and C. List. Judgment aggregation by quota rules: Majority voting generalized. *Journal of Theoretical Politics*, 19(4):391–424, 2007.

[9] F. Dietrich and C. List. Strategy-proof judgment aggregation. *Economics and Philosophy*, 23(3):269–300, 2007.

[10] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[11] J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard–Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

[12] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pages 299–310. Springer-Verlag *Lecture Notes in Computer Science #5814*, October 2009.

[13] U. Endriss, U. Grandi, and D. Porello. Complexity of judgment aggregation: Safety of the agenda. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 359–366. IFAAMAS, May 2010.

[14] U. Endriss, U. Grandi, and D. Porello. Complexity of winner determination and strategic manipulation in judgment aggregation. In V. Conitzer and J. Rothe, editors, *Proceedings of the 3rd International Workshop on Computational Social Choice*, pages 139–150. Universität Düsseldorf, September 2010.

[15] G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, D. Raible, and J. Rothe. The complexity of probabilistic lobbying. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory*, pages 86–97. Springer-Verlag *Lecture Notes in Artificial Intelligence #5783*, October 2009.

[16] G. Erdélyi, L. Piras, and J. Rothe. The complexity of voter partition in Bucklin and fallback voting: Solving three open problems. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 837–844. IFAAMAS, May 2011.

[17] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[18] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[19] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[20] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[21] P. Faliszewski and A. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[22] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[23] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[24] L. Kornhauser and L. Sager. Unpacking the court. *Yale Law Journal*, 96(1):82–117, 1986.

[25] C. List. The theory of judgment aggregation: An introductory review. *Synthese*. To appear.

[26] C. List and C. Puppe. Judgment aggregation: A survey. In P. Anand, P. Pattanaik, and C. Puppe, editors, *Oxford Handbook of Rational and Social Choice*, chapter 19. Oxford University Press, 2009.

[27] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, second edition, 1995.

[28] J. Rothe and L. Schend. Typical-case challenges to complexity shields that are supposed to protect elections against manipulation and control: A survey. In *Website Proceedings of the Special Session on Computational Social Choice at the 12th International Symposium on Artificial Intelligence and Mathematics*, January 2012.

[29] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

Dorothea Baumeister and Jörg Rothe
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: {baumeister,rothe}@cs.uni-duesseldorf.de

Gábor Erdélyi
Institut für Betriebswirtschaftslehre
Universität Siegen
57076 Siegen, Germany
Email: erdelyi@wiwi.uni-siegen.de

Olivia J. Erdélyi
Institut für Versicherungsrecht
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: olivia.erdelyi@uni-duesseldorf.de

The Possible Winner Problem with Uncertain Weights[1]

Dorothea Baumeister, Magnus Roos, Jörg Rothe, Lena Schend, and Lirong Xia

Abstract

The original possible winner problem is: Given an unweighted election with partial preferences and a distinguished candidate *c*, can the preferences be extended to total ones such that *c* wins? We introduce a novel variant of this problem in which not some of the voters' preferences are uncertain but some of their weights. Not much has been known previously about the weighted possible winner problem. We present a general framework to study this problem, both for integer and rational weights, with and without upper bounds on the total weight to be distributed, and with and without ranges to choose the weights from. We study the complexity of these problems for important voting systems such as scoring rules, Copeland, ranked pairs, plurality with runoff, and (simplified) Bucklin and fallback voting.

1 INTRODUCTION

Much of the previous work in computational social choice has focused on the complexity of manipulation, control, and bribery problems in voting (see the surveys by Faliszewski et al. [18, 21]). More recently, many papers studied the possible winner problem, which generalizes the (unweighted) coalitional manipulation problem. The original possible winner problem was introduced by Konczak and Lang [24]. The input to this problem is an election with partial (instead of total) preferences and a distinguished candidate, and the question is whether it is possible to extend the partial preferences to total ones such that the distinguished candidate wins. Xia and Conitzer [28] studied this and also the necessary winner problem. Betzler and Dorn [7] and Baumeister and Rothe [5] established a dichotomy result for the possible winner problem, and Betzler et al. [8, 6] investigated the parameterized complexity of this problem.

A number of variants of the possible winner problem have been studied as well. Bachrach, Betzler, and Faliszewski [1] investigated a probabilistic variant thereof. Chevaleyre et al. [10] introduced the *possible winner with respect to the addition of new alternatives* problem, which is related to, yet different from the problem of control via adding candidates[2](see [2, 23]) and is also similar, yet not identical to the cloning problem in elections [16]. Their variant was further studied by, e.g., Xia, Lang, and Monnot [29] and Baumeister, Roos, and Rothe [4]. The latter paper in particular considered a weighted variant of the possible winner problem, and it also introduced and studied this problem under voting rule uncertainty, an approach that was followed up recently by Elkind and Erdélyi [14] who applied it to coalitional manipulation [11]. Baumeister et al. [3] studied variants of the possible winner problem with truncated ballots. Lang et al. [25] and Pini et al. [27] investigated the possible and necessary winner problem for voting trees and multi-round election systems such as STV. Most of the papers listed above consider only *unweighted* elections. We present a general framework to study the *weighted* possible winner problem, and we focus on elections where not some of the voters' preferences, but some of their *weights*, are uncertain. The problems we study in our framework come with integer or rational weights, with or without upper bounds on the total weight to be assigned, and with or without given ranges to choose the weights from. An interesting point in this regard is that while the original possible winner problem generalizes the coalitional manipulation problem [11], certain variants of the possible winner problem with uncertain weights generalize constructive control by adding/deleting voters [2, 23].

[1]To appear in *Proceedings of the 20th European Conference on Artificial Intelligence*, Montpellier, France, August 2012.
[2]We use *candidate* and *alternative* synonymously.

The following situation may motivate why it is interesting to study the possible winner problem with uncertain weights. Imagine a company that is going to decide on its future strategy by voting at the annual general assembly of stockholders. Among the parties involved, everybody's preferences are common knowledge. However, who will succeed with its preferred alternative for the future company strategy depends on the stockholders' weights, i.e., on how many stocks they each own, and there is uncertainty about these weights. Is it possible to assign weights to the parties involved (e.g., by them buying new stocks) such that a given alternative wins? As another example, suppose we want to decide which university is the best in the world based on different criteria (e.g., graduation and retention rates, faculty resources, student selectivity, etc.). Each criterion can be seen as a voter who gives a ranking over all universities (candidates). Suppose the voting rule is fixed (e.g., plurality), but the chair can determine the weights of these criteria. It is interesting to know whether a given university can win if the chair chooses the weights carefully.

2 PRELIMINARIES

An *election* is a pair (C,V) consisting of a finite set C of candidates and a finite list V of voters that are represented by their preferences over the candidates in C and are occasionally denoted by $v_1,\dots,v_{|V|}$. A voting system $\mathscr{E}$ is a set of rules determining the winning candidates according to the preferences in V. The voting systems considered here are all preference-based, that is, the votes are given as linear orders over C. For example, if $C=\{a,b,c,d\}$ then a vote $a>c>b>d$ means that this voter (strictly) prefers a to c, c to b, and b to d. If such an order is not total (e.g., when a voter only specifies $a>c>d$ as her preference over these four candidates), we say it is a partial order. For winner determination in weighted voting systems, a vote v of weight w is considered as if there were w unweighted (i.e., unit-weight) votes v.

For a given election (C,V), the *weighted majority graph (WMG)* is defined as a directed graph whose vertices are the candidates, and we have an edge $c\to d$ of weight $N(c,d)$ between any two vertices c and d, where $N(c,d)$ is the number of voters preferring c to d minus the number of voters preferring d to c. Note that in the WMG of any election, all weights on the edges have the same parity (and whether it is odd or even depends on the parity of the number of votes), and $N(c,d)=-N(d,c)$ (which is why it is enough to give only one of these two edges explicitly).

We will consider the following voting rules.

- **Positional Scoring Rules:** These rules are defined by a *scoring vector* $\vec{\alpha}=(\alpha_1,\alpha_2,\dots,\alpha_m)$, where m is the number of candidates, the α_i are nonnegative integers, and $\alpha_1\geq\alpha_2\geq\cdots\geq\alpha_m$. Let $\rho_i(c)$ denote the position of candidate c in voter v_i's vote. Then c receives $\alpha_{\rho_i(c)}$ points from v_i, and the total score of c is $\sum_{i=1}^{n}\alpha_{\rho_i(c)}$ for n voters. All candidates with the largest score are the $\vec{\alpha}$ winners. In particular, we will consider k-approval elections, $k\leq m$, whose scoring vector has a 1 in the first k positions, and the remaining $m-k$ entries are all 0. The special case of 1-approval is also known as *plurality* and that of $(m-1)$-approval as *veto*. The scoring vector $(m-1,m-2,\dots,2,1,0)$ defines the *Borda* rule.

- **Copeland$^\alpha$ (for each rational number α, $0\leq\alpha\leq 1$):**[3] For any two alternatives c and c', we can simulate a *pairwise election* between them, by seeing how many voters prefer c to c', and how many prefer c' to c; the winner of the pairwise election is the one preferred more often. Then, an alternative receives one point for each win in a pairwise election, α points for each tie, and zero points for each loss. This is the Copeland score of the alternative. A Copeland winner maximizes the Copeland score.

- **Ranked pairs:** This rule first creates an entire ranking of all the candidates. In each step, we consider a pair of candidates c,c' that we have not previously considered; specifically, we

[3] The original Copeland system [12] is defined for the specific value of $\alpha=1/2$; the generalization to other α values is due to Faliszewski et al. [20].

choose among the remaining pairs one with the highest $N(c,c')$ value (in case of ties, we use some tie-breaking mechanism) and then fix the order $c > c'$, unless this contradicts previous orders already fixed (i.e., unless this order violates transitivity). We continue until we have considered all pairs of candidates (and so we have a full ranking). A candidate at the top of the ranking for some tie-breaking mechanism is a winner.

- **Plurality with runoff:** This rule proceeds in two rounds. First, all alternatives except those two with the highest plurality score are eliminated; in the second round (the runoff), the plurality rule is used to select a winner among these two. Some tie-breaking rule is applied in both rounds if needed.
- **Bucklin and fallback voting (both simplified):** In a Bucklin election, the voters' preferences are linear orders and the level ℓ score of a candidate c is the number of voters ranking c among their top ℓ positions. The Bucklin score of a candidate c is the smallest number t such that more than half of the voters rank c somewhere in their top t positions. A Bucklin winner minimizes the Bucklin score.[4] In (simplified) fallback elections, on the other hand, nontotal (more specifically, "top-truncated" as defined in [3]) preference orders are allowed. Every Bucklin winner is also a fallback winner, but if no Bucklin winner exists (which may happen due to the voters' partial orders) and ℓ is the length of a longest preference order among the votes, all candidates with the greatest level ℓ score are the fallback winners. Throughout this paper we will refer to "simplified Bucklin" and "simplified fallback" simply as Bucklin and fallback voting.

We will use the following notation. If the set of candidates is, say, $C = B \cup D \cup \{c\}$, then we mean by $c > \overrightarrow{D} > \cdots$ that c is preferred to all candidates, where $\overrightarrow{D}$ is an arbitrarily fixed ordering of the candidates occurring in D, and "$\cdots$" indicates that the remaining candidates (those from B in this example) can be ranked in an arbitrary order afterwards.

Some proofs in this paper use *McGarvey's trick* [26] (applied to WMGs), which constructs a list of votes whose WMG is the same as some targeted weighted directed graph. This will be helpful because when we present our proofs, we only need to specify the WMG instead of the whole list of votes, and then by using McGarvey's trick for WMGs, a votes list can be constructed in polynomial time. More specifically, McGarvey showed that for every unweighted majority graph, there is a particular list of preferences that results in this majority graph. Extending this to WMGs, the trick works as follows. For any pair of candidates, (c,d), if we add two votes, $c > d > c_3 > \cdots > c_m$ and $c_m > c_{m-1} > \cdots > c_3 > c > d$, to a vote list, then in the WMG, the weight on the edge $c \to d$ is increased by 2 and the weight on the edge $d \to c$ is decreased by 2, while the weights on all other edges remain unchanged.

3 PROBLEM DEFINITIONS AND DISCUSSION

We now define our variants of the possible winner problem with uncertain weights. Let $\mathscr{E}$ be a given voting system and $\mathbb{F} \in \{\mathbb{Q}^+, \mathbb{N}\}$.

$\mathscr{E}$-Possible-Winner-with-Uncertain-Weights-$\mathbb{F}$ ($\mathscr{E}$-PWUW-$\mathbb{F}$)	
Given:	An $\mathscr{E}$ election $(C, V_0 \cup V_1)$, $V_0 \cap V_1 = \emptyset$, where the weights of the voters in V_0 are not specified yet and weight zero is allowed for them, yet all voters in V_1 have weight one, and a designated candidate $c \in C$.
Question:	Is there an assignment of weights $w_i \in \mathbb{F}$ to the votes v_i in V_0 such that c is an $\mathscr{E}$ winner of election $(C, V_0 \cup V_1)$ when v_i's weight is w_i for $1 \le i \le \|V_0\|$?

[4] We consider only this simplified version of Bucklin voting. In the full version (see, e.g., [17]), among all candidates with smallest Bucklin score, say t, for c to win it is also required that c's level t score is largest.

We distinguish between allowing nonnegative rational weights (i.e., weights in $\mathbb{Q}^+$) and nonnegative integer weights (i.e., weights in $\mathbb{N}$). In particular, we allow weight-zero voters in V_0. Note that for inputs with $V_0 = \emptyset$ (which is not excluded in the problem definition), we obtain the ordinary unweighted (i.e., unit-weight) winner problem for $\mathscr{E}$. Allowing weight zero for voters in V_0 in some sense corresponds to control by deleting voters (see [2, 23]); later in this section we also briefly discuss the relationship with control by adding voters. The reason why we distinguish between votes with uncertain weights and unit-weight votes in our problem instances is that we want to capture these problems in their full generality; just as votes with total preferences are allowed to occur in the instances of the original possible winner problem. The requirement of normalizing the weights in V_1 to unit-weight, on the other hand, *is* a restriction (that doesn't hurt) and is chosen at will. This will somewhat simplify our proofs.

We also consider the following restrictions of $\mathscr{E}$-PWUW-$\mathbb{F}$:

- In $\mathscr{E}$-PWUW-RW-$\mathbb{F}$, an $\mathscr{E}$-PWUW-$\mathbb{F}$ instance and regions (i.e., intervals) $R_i \subseteq \mathbb{F}$, $1 \leq i \leq |V_0|$, are given, and the question is the same as in $\mathscr{E}$-PWUW-$\mathbb{F}$, except that each weight w_i must be chosen from R_i in addition.
- In $\mathscr{E}$-PWUW-BW-$\mathbb{F}$, an $\mathscr{E}$-PWUW-$\mathbb{F}$ instance and a positive bound $B \in \mathbb{F}$ is given, and the question is the same as in $\mathscr{E}$-PWUW-$\mathbb{F}$, except that $\sum_{i=1}^{|V_0|} w_i \leq B$ must hold in addition (i.e., the total weight that can be assigned must be bounded by B).
- In $\mathscr{E}$-PWUW-BW-RW-$\mathbb{F}$, an $\mathscr{E}$-PWUW-BW-$\mathbb{F}$ instance and regions (i.e., intervals) $R_i \subseteq \mathbb{F}$, $1 \leq i \leq |V_0|$, are given, and the question is the same as in $\mathscr{E}$-PWUW-BW-$\mathbb{F}$, except that each weight w_i must be chosen from R_i in addition.

One could also define other variants of $\mathscr{E}$-PWUW-$\mathbb{F}$ (e.g., the *destructive* variant where the question is whether c's victory can be prevented by some weight assignment) or other variants of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{F}$ and $\mathscr{E}$-PWUW-RW-$\mathbb{F}$ (e.g., by allowing *sets of intervals* for each weight), but here we focus on the eight problems defined above. We focus on the *winner* model (aka. the *co-winner* or the *nonunique-winner* model) where the question is whether c can be made *a* winner by assigning appropriate weights. By minor proof adjustments, most of our results can be shown to also hold in the *unique-winner* model where we ask whether c can be made the only winner.

We assume that the reader is familiar with common complexity-theoretic notions, such as the complexity classes P and NP, and the notions of hardness and completeness with respect to the polynomial-time many-one reducibility, which we denote by $\leq_m^p$.

The following reductions hold trivially among our problems, by setting the bound on the total weight allowed to the sum of the highest possible weights for the first two reductions and by setting the intervals to $[0,B]$ (where B is the bound on the total weight) for the last two reductions:

$$\begin{aligned} \text{PWUW-RW-}\mathbb{Q}^+ &\leq_m^p \text{PWUW-BW-RW-}\mathbb{Q}^+ && (1)\\ \text{PWUW-RW-}\mathbb{N} &\leq_m^p \text{PWUW-BW-RW-}\mathbb{N} && (2)\\ \text{PWUW-BW-}\mathbb{Q}^+ &\leq_m^p \text{PWUW-BW-RW-}\mathbb{Q}^+ && (3)\\ \text{PWUW-BW-}\mathbb{N} &\leq_m^p \text{PWUW-BW-RW-}\mathbb{N}. && (4) \end{aligned}$$

Related to our variants of the PWUW problem is the problem of constructive control by adding voters (see [2]), CCAV for short. Here, a set C of candidates with a distinguished candidate $c \in C$, a list V of registered voters, an additional list V' of as yet unregistered voters, and a positive integer k are given. The question is whether it is possible to make c win the election by adding at most k voters from V' to the election.

Obviously, there is a direct polynomial-time many-one reduction from CCAV to PWUW-BW-RW-$\mathbb{N}$. The voters in V_1 are the registered voters from V and the voters in V_0 are those from V', where the weights can be chosen from $\{0,1\}$ for all votes in V_0, and the total bound on the weight B is set

PWUW-	Scoring Rules, Plurality with runoff	Plurality, 2-AV, Veto	3-AV	k-AV, $k \geq 4$	Bucklin, Fallback	Copeland, Ranked Pairs
$\mathbb{Q}^+$	P	P	P	P	P	?
$\mathbb{N}$	?	P	P	P	NP-c.	NP-c.
BW-RW-$\mathbb{Q}^+$	P	P	P	P	P	?
BW-RW-$\mathbb{N}$	?	P	?	NP-c.	NP-c.	NP-c.
BW-$\mathbb{Q}^+$	P	P	P	P	P	?
BW-$\mathbb{N}$	?	P	?	NP-c.	NP-c.	NP-c.
RW-$\mathbb{Q}^+$	P	P	P	P	P	?
RW-$\mathbb{N}$	?	P	P	P	NP-c.	NP-c.

Table 1: Overview of results. "NP-c." stands for NP-complete.

to k. If succinct representation is assumed,[5] there is also a polynomial-time many-one reduction in the other direction. The registered voters are those from V_1, and the unregistered voters are those from V_0, where each vote is added according to its maximal weight in the PWUW instance. The number k of voters who may be added equals the bound B on the total weight.

Since there are reductions in both directions, complexity results carry over from CCAV to PWUW-BW-RW-$\mathbb{N}$ when we assume succinct representation. For the voting systems considered in this paper, this implies that PWUW-BW-RW-$\mathbb{N}$ is NP-complete for Copeland^0 and Copeland^1, and is solvable in polynomial time for plurality (see [20, 2]). (Note that the NP-hardness results on CCAV for Bucklin and fallback voting from [17] concern the full, not the simplified versions of these voting rules.) These already known cases are nevertheless covered by our proofs in the next section, since they handle several restrictions of the PWUW problems at the same time. Conversely, the results from the next section for PWUW-BW-RW-$\mathbb{N}$ all carry over to CCAV if we assume succinct representation.

4 RESULTS AND SELECTED PROOFS

Table 1 gives an overview of our results. In the next section, we will provide or sketch some of the proofs for these results. Due to space constraints, not all proofs can be presented in full detail.

4.1 Integer Weights

We begin with the results for the integer cases.

Proposition 1 *1. Each of the four variants of* plurality-PWUW-$\mathbb{N}$, veto-PWUW-$\mathbb{N}$, *and* 2-approval-PWUW-$\mathbb{N}$ *studied in this paper is in* P.

2. For each $k \geq 1$, k-approval-PWUW-$\mathbb{N}$ *and* k-approval-PWUW-RW-$\mathbb{N}$ *are in* P.

PROOF. For the first statement, we present the proof details for 2-approval-PWUW-BW-RW-$\mathbb{N}$, where for each vote in V_0 the range of allowed weights is $\{0,1\}$. The proof can be adjusted to also work when other ranges are given.

Given a 2-approval-PWUW-BW-RW-$\mathbb{N}$ instance as above, we construct the following max-flow instance. Let V_0' denote the list of votes in V_0 where c is ranked among the top two positions. We may assume, without loss of generality, that the given bound B on the total weight satisfies $B \leq |V_0'|$.[6] The vertices are $\{s, s', t\} \cup V_0' \cup (C \setminus \{c\})$ with the following edges:

[5] This means that when there are several identical votes, we don't list them all but rather store a number in binary saying how often this vote occurs.

[6] Otherwise, the optimal strategy is to let the weights of the votes in V_0' be 1 and to let the weights of all other votes be 0.

- There is an edge $s \to s'$ with capacity B and an edge from s' to each node in V_0' with capacity 1.
- There is an edge from a node L in V_0' to a node d in $C \setminus \{c\}$ with capacity 1 if and only if d is ranked besides c among the top two positions in L.
- There is an edge from each node $d \in C \setminus \{c\}$ to t with capacity $B + score(c, V_1) - score(d, V_1)$, where $score(e, V_1)$ is the 2-approval score of any $e \in C$ in vote list V_1.[7]

In the max-flow problem, we are asked whether there exists a flow whose value is B. We note that in the PWUW instance, it is always optimal to choose B votes in V_0' and to let their weights be 1. The bound on $d \to t$ for $d \in C \setminus \{c\}$ ensures that the 2-approval score of d is no more than the 2-approval score of c.

The claims for 2-approval-PWUW-RW-$\mathbb{N}$ and 2-approval-PWUW-BW-$\mathbb{N}$ follow from (2) and (4).

For the second statement, it suffices to maximize the weights of the votes in V_0' that rank c among their top k positions, and to minimize the weights of the other votes.

❑

In particular, it is open whether 3-approval-PWUW-BW-RW-$\mathbb{N}$ and 3-approval-PWUW-BW-$\mathbb{N}$ are also in P. For $k \geq 4$, however, we can show that these problems are NP-complete.

Theorem 2 *For each $k \geq 4$,* k-approval-PWUW-BW-RW-$\mathbb{N}$ *and* k-approval-PWUW-BW-$\mathbb{N}$ *are* NP-*complete.*

PROOF. It is easy to see that both problems belong to NP. For proving NP-hardness, we give a proof for 4-approval-PWUW-BW-$\mathbb{N}$ by a reduction from the NP-complete problem EXACT COVER BY 3-SETS (X3C): Given a set $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and a collection $\mathscr{S} = \{S_1, \ldots, S_n\}$ with $|S_i| = 3$ and $S_i \subseteq \mathscr{B}$, $1 \leq i \leq n$, does $\mathscr{S}$ contain an exact cover for $\mathscr{B}$, i.e., a subcollection $\mathscr{S}' \subseteq \mathscr{S}$ such that every element of $\mathscr{B}$ occurs in exactly one member of $\mathscr{S}'$?

Construct an instance of k-approval-PWUW-BW-$\mathbb{N}$ with the set

$$C = \{c, b_1, \ldots, b_{3q}, b_1^1, \ldots, b_{3q}^1, b_1^2, \ldots, b_{3q}^2, b_1^3, \ldots, b_{3q}^3\}$$

of candidates, where c is the designated candidate, and with the set V_0 of n votes of the form $c > \overrightarrow{S_i} > \cdots$, the set V_1 of $q-1$ votes of the form $b_j > b_j^1 > b_j^2 > b_j^3 > \cdots$ for each j, $1 \leq j \leq 3q$, and the bound $B = q$ on the total weight of the votes in V_0. Recall that the votes in V_1 all have fixed weight one, and those of the votes in V_0 are from $\mathbb{N}$. We show that $\mathscr{S}$ has an exact cover for $\mathscr{B}$ if and only if we can set the weights of the voters in this election such that c is a winner.

Assume that there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ for $\mathscr{B}$. By setting the weights of the votes $c > \overrightarrow{S_i} > \cdots$ to one for those q subsets S_i contained in $\mathscr{S}'$, and to zero for all other votes in V_0, c is a winner of the election, as c and all b_j, $1 \leq j \leq 3q$, receive exactly q points, whereas b_j^1, b_j^2, and b_j^3, $1 \leq j \leq 3q$, receive $q-1$ points each.

Conversely, assume that c can be made a winner of the election by choosing the weights of the votes in V_0 appropriately. Note that the bound on the total weight for the votes in V_0 is $B = q$. Every b_i gets $q-1$ points from the votes in V_1, and c gets points only from the votes in V_0. Since there are always some b_j getting points if a vote from V_0 has weight one, there are at least three b_j having q points if a vote from V_0 has weight one. Hence c must get q points from the votes in V_0 by setting the weight of q votes to one. Furthermore, every b_j can occur only once in the votes having weight one in V_0, as otherwise c would not win. Thus, the S_i corresponding to the votes of weight one in V_0 must form an exact cover for $\mathscr{B}$.

[7]Note that if this capacity is negative, the given 2-approval-PWUW-BW-RW-$\mathbb{N}$ instance is trivially a no-instance, since c can never be made a winner.

By adding dummy candidates to fill the positions receiving points, we can adapt this proof for k-approval for any fixed $k > 4$. NP-hardness for k-approval-PWUW-BW-RW-$\mathbb{N}$, $k \geq 4$, then follows from the trivial reduction (4) stated in Section 3. ❑

We now show that all variants of PWUW with integer weights are NP-complete for Copeland$^{\alpha}$, ranked pairs, Bucklin, and fallback elections.

Theorem 3 *For each rational number α, $0 \leq \alpha \leq 1$, every variant of* Copeland$^{\alpha}$-PWUW-$\mathbb{N}$ *studied in this paper is* NP-*complete.*

PROOF. NP membership is easy to see for all problem variants. We first prove NP-hardness for Copeland$^{\alpha}$-PWUW-$\mathbb{N}$, and then show how to modify the proof for the variants of the problem. Given an X3C instance $(\mathscr{B}, \mathscr{S})$ with $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathscr{S} = \{S_1, \ldots, S_n\}$, we construct the following PWUW instance for Copeland$^{\alpha}$, where the set of candidates is $\mathscr{B} \cup \{c, d, e\}$. Without loss of generality we assume that $q \geq 4$ and we are asked whether c can be made a winner.

The votes on C are defined as follows. V_0 will encode the X3C instance and V_1 will be used to implement McGarvey's trick. V_0 consists of the following n votes: For each j, $1 \leq j \leq n$, there is a vote $d > e > \overrightarrow{S_j} > c > \cdots$. V_1 is the vote list whose WMG has the following edges:

- $c \to d$ with weight $q+1$, $d \to e$ with weight $q+1$, and $e \to c$ with weight $q+1$.
- For every i, $1 \leq i \leq 3q$, $d \to b_i$ and $e \to b_i$ each with weight $q+1$, and $b_i \to c$ with weight $q-3$.
- The weight on any other edge not defined above is no more than 1.

It follows that no matter what the weights of the votes in V_0 are, d beats e and e beats c in pairwise elections, and both d and e beat all candidates in $\mathscr{B}$ in pairwise elections. For c to be a winner, c must beat d in their pairwise election, which means that the total weight of the votes in V_0 is no more than q. On the other hand, c must beat all candidates in $\mathscr{B}$. This happens if and only if the votes in V_0 that have positive weights correspond to an exact cover of $\mathscr{B}$, and all of these votes must have weight one. This means that Copeland$^{\alpha}$-PWUW-$\mathbb{N}$ is NP-hard.

For the BW and BW-RW variants, we let $B = q$; for the RW and BW-RW variants, we let the range of each vote in V_0 be $\{0, 1\}$. ❑

Theorem 4 *All variants of* ranked-pairs-PWUW-$\mathbb{N}$ *studied in this paper are* NP-*complete.*

PROOF. The proof is similar to the proof of Theorem 3. That the problems are in NP is easy to see. For the hardness proof, given an X3C instance $(\mathscr{B}, \mathscr{S})$ with $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathscr{S} = \{S_1, \ldots, S_n\}$, we construct the following ranked-pairs-PWUW-$\mathbb{N}$ instance, where the set of candidates is $\mathscr{B} \cup \{c, d\}$. We are asked whether c can be made a winner. V_0 consists of the following n votes: For each $j, 1 \leq j \leq n$, there is a vote $e > \overrightarrow{S_j} > c > d > \cdots$. V_1 is the vote list whose WMG has the following edges, and is constructed by applying McGarvey's trick:

- $c \to d$ with weight $2q+1$, $d \to e$ with weight $4q+1$, and $e \to c$ with weight $2q+1$.
- For every i, $1 \leq i \leq 3q$, $d \to b_i$ and $e \to b_i$ each with weight $2q+1$, and $b_i \to c$ with weight $4q-1$.
- The weight on any other edge not defined above is 1.

If the total weight of votes in V_0 is larger than q, then the weight on $e \to c$ and $e \to b_i$ in the WMG is at least $3q+2$, and the weight on $d \to e$ is no more than $3q$, which means that c is not a winner for ranked pairs. Moreover, if c is a winner, then the weight on any $b_i \to c$ should not be strictly higher than the weight on $c \to d$, otherwise $b_i \to c$ will be fixed in the final ranking. It follows that if c is a winner, then the votes in V_0 that have positive weights correspond to an exact cover of $\mathscr{B}$, and all of these votes must have weight one. This means that ranked-pairs-PWUW-$\mathbb{N}$ is NP-hard.

For the BW and BW-RW variants, we let $B = q$; for the RW and BW-RW variants, we let the range of each vote in V_0 be $\{0,1\}$. ❑

Theorem 5 *All variants of* Bucklin-PWUW-$\mathbb{N}$ *studied in this paper are* NP-*complete.*

PROOF. NP membership is easy to see for all problem variants. We first prove NP-hardness for Bucklin-PWUW-$\mathbb{N}$, and then show how to modify the proof for the variants of the problem. Given an X3C instance $(\mathscr{B},\mathscr{S})$ with $\mathscr{B} = \{b_1,\dots,b_{3q}\}$ and $\mathscr{S} = \{S_1,\dots,S_n\}$, we construct the following Bucklin-PWUW-$\mathbb{N}$ instance. The set of candidates is $\mathscr{B} \cup \{c,d\} \cup D \cup D'$, where $D = \{d_1,\dots,d_{3q}\}$ and $D' = \{d'_1,\dots,d'_{3q}\}$ are sets of auxiliary candidates. We are asked whether c can be made a winner. V_0 consists of the following n votes: For each j, $1 \leq j \leq n$, there is a vote $d > \overrightarrow{S_j} > c > \overrightarrow{D} > \overrightarrow{D'} > \cdots$. V_1 consists of $q-1$ copies of $\overrightarrow{\mathscr{B}} > c > \overrightarrow{D'} > \overrightarrow{D} > d$ and one copy of $\overrightarrow{D'} > c > \overrightarrow{\mathscr{B}} > d > \overrightarrow{D}$.

If the total weight of votes in V_0 is larger than q, then d is the unique candidate that is ranked in top positions for more than half of the votes, which means that c is not a winner. Suppose the total weight of the votes in V_0 is at most q. Then, the Bucklin score of c is $3q+1$ and the Bucklin score of any candidate in D and D' is larger than $3q+1$. Therefore, c is a Bucklin winner if and only if the Bucklin score of any candidate in $\mathscr{B}$ is at least $3q+1$. This happens if and only if the votes in V_0 that have positive weights correspond to an exact cover of $\mathscr{B}$, and all of these votes must have weight one. This means that Bucklin-PWUW-$\mathbb{N}$ is NP-hard.

For the BW and BW-RW variants, we let $B = q$; for the RW and BW-RW variants, we let the range of each vote in V_0 be $\{0,1\}$. ❑

Bucklin voting can be seen as the special case of fallback voting where all voters give complete linear orders over all candidates. So the NP-hardness results for Bucklin voting transfer to fallback voting, while the upper NP bounds are still easy to see.

Corollary 6 *All variants of* fallback-PWUW-$\mathbb{N}$ *studied in this paper are* NP-*complete.*

4.2 Rational Weights and Voting Systems that Can Be Represented by Linear Inequalities

Chamberlin and Cohen [9] observed that various voting rules can be represented by systems of linear inequalitites, see also [19]. We use this property to formulate linear programs, thus being able to solve the PWUW problem variants with rational weights for these voting rules efficiently, provided that the size of the systems describing the voting rules is polynomially bounded. Note that an LP with rational instead of integer values can be solved in polynomial time [22].

What voting rules does this technique apply to? The crucial requirement a voting rule needs to satisfy is that the scoring function used for winner determination can be described by linear inequalities and that this description is in a certain sense independent of the voters' weights. By "independent of the voters' weights" we mean that the points a candidate gains from a vote are determined essentially in the same way in both a weighted and an unweighted electorate, but in the former we have a weighted sum of these points that gives the candidate's score, whereas in the latter we have a plain sum. Scoring functions satisfying this condition are said to be *weight-independent*. This requirement is fulfilled by, e.g., the scoring functions of all scoring rules, Bucklin, and fallback voting. Copeland's scoring function, on the other hand, does not satisfy it. In a Copeland election, every candidate gets one point for each other candidate she beats in a pairwise contest. Who of the two candidates wins a pairwise contest and thus gains a Copeland point depends directly on the voters' weights. Thus, the Copeland score in a weighted election is not a weighted sum of the Copeland scores in the corresponding unweighted election in the above sense.

In what follows, we have elections where the voter list consists of the two sublists V_0 and V_1. We have to assign weights $x_1,\dots,x_{|V_0|}$ to the voters in V_0. We don't exclude the case where weight zero can be assigned, but we will seek to find solutions where all weights are strictly positive, since assigning weight zero to a voter is equivalent to excluding this voter entirely from the election. For $c \in C$, let $\rho_i^0(c)$ denote the position of c in the preference of the ith voter in V_0, $1 \leq i \leq |V_0|$, and let $\rho_j^1(c)$ denote the position of c in the preference of the jth voter in V_1, $1 \leq j \leq |V_1|$.

Lemma 7 *Let $\mathscr{E}$ be a voting rule with a weight-independent scoring function that can be described by a system A of polynomially many linear inequalities. Then* $\mathscr{E}$-PWUW-$\mathbb{Q}^+$, $\mathscr{E}$-PWUW-BW-$\mathbb{Q}^+$, $\mathscr{E}$-PWUW-RW-$\mathbb{Q}^+$, *and* $\mathscr{E}$-PWUW-BW-RW-$\mathbb{Q}^+$ *are each in* P.

PROOF. Let $x_1,x_2,\dots,x_n$ be the variables of the system A that describes $\mathscr{E}$ for an $\mathscr{E}$ election with n voters. The following linear program can be used to solve $\mathscr{E}$-PWUW-BW-RW-$\mathbb{Q}^+$. Let an instance of this problem be given: an election $(C, V_0 \cup V_1)$ with as yet unspecified weights in V_0, a designated candidate $c \in C$, a bound $B \in \mathbb{Q}^+$, and regions $R_i \subseteq \mathbb{Q}^+$, $1 \leq i \leq |V_0|$. The *vector of variables* of our linear program is $\vec{x} = (x_1,x_2,\dots,x_{|V_0|},\chi) \in \mathbb{R}^{|V_0|+1}$ and we maximize the *objective function* $\vec{c}\cdot\vec{x}^T$ with $\vec{c} = (0,0,\dots,0,1)$ and the following constraints:

$$A \tag{5}$$
$$x_i - \chi \geq 0 \qquad \text{for } 1 \leq i \leq |V_0| \tag{6}$$
$$\chi \geq 0 \tag{7}$$
$$\sum_{i=1}^{|V_0|} x_i \leq B \tag{8}$$
$$x_i \leq r_i \qquad \text{for } 1 \leq i \leq |V_0| \tag{9}$$
$$-x_i \leq -\ell_i \quad \text{for } 1 \leq i \leq |V_0| \tag{10}$$

Constraint (5) gives the linear inequalitites that have to be fulfilled for the designated candidate c to win under $\mathscr{E}$. By maximizing the additional variable χ in the objective function we try to find solutions where the weights are positive, this is accomplished by constraint (6). Constraint (8) implements our given upper bound B for the total weight to be assigned and constraints (9) and (10) implement our given ranges $R_i = [\ell_i, r_i] \subseteq \mathbb{Q}$ for each weight.

Omit (8) for $\mathscr{E}$-PWUW-RW-$\mathbb{Q}^+$, omit (9) and (10) for $\mathscr{E}$-PWUW-BW-$\mathbb{Q}^+$, and omit (8), (9), and (10) for $\mathscr{E}$-PWUW-$\mathbb{Q}^+$.

A solution in $\mathbb{Q}$ for a linear program with polynomially bounded constraints can be found in polynomial time. ❑

In the following theorems we present the specific systems of linear inequalities describing scoring rules in general, and the voting systems Bucklin, fallback, and plurality with runoff. These can be used to formally specify the complete linear program stated in the proof of Lemma 7.

Theorem 8 *For each scoring rule $\vec{\alpha}$,* $\vec{\alpha}$-PWUW-$\mathbb{Q}^+$, $\vec{\alpha}$-PWUW-BW-$\mathbb{Q}^+$, $\vec{\alpha}$-PWUW-RW-$\mathbb{Q}^+$, *and* $\vec{\alpha}$-PWUW-BW-RW-$\mathbb{Q}^+$ *are in* P.

PROOF. We are given an election with m different candidates in C, where $c \in C$ is the distinguished candidate. Recall that $\rho_i^0(c)$ denotes c's position in the preference of voter $v_i \in V_0$, and that $\alpha_{\rho_i^0(c)}$ denotes the number of points c gets for this position according to the scoring vector $\vec{\alpha}$. Let $S_{V_1}(c)$ denote the number of points candidate c gains from the voters in V_1 (recall that those have all weight one). Then the distinguished candidate c is a winner if and only if for all candidates $c' \in C$ with

$c' \neq c$, we have $\left(\left(\alpha_{\rho_j^0(c)} - \alpha_{\rho_j^0(c')}\right)_{1\leq j\leq |V_0|}\right)\vec{x}^T \geq S_{V_1}(c') - S_{V_1}(c)$, where $\vec{x} = (x_1, x_2, \ldots, x_{|V_0|}) \in \mathbb{R}^{|V_0|}$ are the weights that will be assigned to the voters in V_0. The linear program for scoring rule $\vec{\alpha}$ is of the following form. As in the proof of Lemma 7, we have the vector of variables $\vec{x} = (x_1, x_2, \ldots, x_{|V_0|}, \chi) \in \mathbb{R}^{|V_0|+1}$ and we maximize the objective function $\vec{c} \cdot \vec{x}^T$ with $\vec{c} = (0, 0, \ldots, 0, 1)$ and the following constraints:

$$-\sum_{i=1}^{|V_0|}\left(\alpha_{\rho_i^0(c)} - \alpha_{\rho_i^0(c')}\right)x_i \leq S_{V_1}(c) - S_{V_1}(c') \;\forall\, c' \neq c \tag{11}$$

$$x_i - \chi \geq 0 \qquad \text{for } 1 \leq i \leq |V_0| \tag{12}$$

$$\chi \geq 0 \tag{13}$$

$$\sum_{i=1}^{|V_0|} x_i \leq B \tag{14}$$

$$x_i \leq r_i \qquad \text{for } 1 \leq i \leq |V_0| \tag{15}$$

$$-x_i \leq -\ell_i \quad \text{for } 1 \leq i \leq |V_0| \tag{16}$$

Here again, constraints (14) to (16) are needed only for the restricted variants.

Since we have at most $(m-1)|V_0| + 3|V_0| + 2 = (m+2)|V_0| + 2$ constraints, this linear program can be solved in polynomial time. ❑

Note that by adding χ to the left-hand side of (11), a solution where χ is positive is an assignment of weights making the distinguished candidate a unique winner.

Being level-based voting rules, for Bucklin and fallback voting we have to slightly expand the presented approach. Due to space constraints, we omit the proof of Theorem 9 and only briefly sketch the idea. Intuitively, it is clear that we first try to make the distinguished candidate a level 1 winner; if this attempt fails, we try the second level; and so on. So the linear program in the proof of Theorem 9 has to be solved for each level beginning with the first until a solution has been found. For Bucklin voting, the representation by linear inequalities is due to Dorn and Schlotter [13], and we adapt it for the simplified version of Bucklin and fallback voting. For the latter, we add appropriate constraints if the approval stage is reached.

Theorem 9 *Let $\mathscr{E}$ be either Bucklin or fallback voting.* $\mathscr{E}$-PWUW-$\mathbb{Q}^+$, $\mathscr{E}$-PWUW-BW-$\mathbb{Q}^+$, $\mathscr{E}$-PWUW-RW-$\mathbb{Q}^+$, *and* $\mathscr{E}$-PWUW-BW-RW-$\mathbb{Q}^+$ *are each in* P.

Note that the proof of Theorem 9 does not work in the unique-winner case.

For plurality with runoff we can take a similar approach: For each candidate d different from c, we use a set of linear inequalities to figure out whether there exists a set of weights such that (1) c and d enter the runoff (i.e., the plurality scores of c and d are at least the plurality score of any other candidate), and (2) c beats d in their pairwise election. Therefore, we have the following corollary whose proof does not work in the unique-winner case.

Theorem 10 *Let PR be the plurality with runoff rule.* PR-PWUW-$\mathbb{Q}^+$, PR-PWUW-BW-$\mathbb{Q}^+$, PR-PWUW-RW-$\mathbb{Q}^+$, *and* PR-PWUW-BW-RW-$\mathbb{Q}^+$ *are each in* P.

PROOF. For each candidate d different from c, there exists a set of linear inequalities that are similar to those in the proof of Theorem 8 such that c and d enter the runoff if and only if these inequalities can be satisfied. We also add the following inequality: $\sum_{\{i \mid c >_{\rho_{0,i}} d\}} x_i + |\{k \mid c >_{\rho_{1,k}} d\}| \geq \sum_{\{i \mid d >_{\rho_{0,i}} c\}} x_i + |\{k \mid d >_{\rho_{1,k}} c\}|$, where $\{i \mid c >_{\rho_{j,i}} d\}$ denotes those voters $v_i \in V_j$ for $j \in \{0,1\}$ that prefer c to d. Then, for each candidate d different from c we construct an LP that is similar to the LP in the proof of Theorem 8. It follows that c is a possible winner if and only if at least one of these LPs has a feasible solution. ❑

5 CONCLUSIONS AND OPEN QUESTIONS

We introduced the possible winner problem with uncertain weights, where not the preferences but the weights of the voters are uncertain, and we studied this problem and its variants in a general framework. We showed that some of these problem variants are easy to solve and some are hard to solve for some of the most important voting rules. Interestingly, while the original possible winner problem (in which there is uncertainty about the voters' preferences) generalizes the coalitional manipulation problem and is a special case of swap bribery [15], the possible winner problem with uncertain weights generalizes the problem of constructive control by adding or deleting voters.

Some interesting issues remain open, as indicated in Table 1, e.g., regarding 3-approval, Copeland voting, positional scoring rules, and plurality with runoff. Also, it would be interesting to study an even more general variant: the weighted possible winner problem with uncertainty about both the voters' preferences and their weights.

Acknowledgments: This work was supported in part by DFG grant RO-1202/15-1, SFF grant "Cooperative Normsetting" of HHU Düsseldorf, ARC-DP110101792, and a DAAD grant for a PPP-PROCOPE project. Lirong Xia is supported by NSF under Grant #1136996 to the Computing Research Association for the CIFellows Project.

References

[1] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 697–702. AAAI Press, July 2010.

[2] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8/9):27–40, 1992.

[3] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. Campaigns for lazy voters: Truncated ballots. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 577–584. IFAAMAS, June 2012.

[4] D. Baumeister, M. Roos, and J. Rothe. Computational complexity of two variants of the possible winner problem. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 853–860. IFAAMAS, May 2011.

[5] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. *Information Processing Letters*, 112(5):186–190, 2012.

[6] N. Betzler. On problem kernels for possible winner determination under the *k*-approval protocol. In *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science*, pages 114–125. Springer-Verlag *Lecture Notes in Computer Science #6281*, August 2010.

[7] N. Betzler and B. Dorn. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *Journal of Computer and System Sciences*, 76(8):812–836, 2010.

[8] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 53–58. IJCAI, July 2009.

[9] J. Chamberlin and M. Cohen. A linear inequality method of establishing certain social choice conjectures. *Public Choice*, 33(2):5–16, 1978.

[10] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. *Mathematical Social Sciences*, 64(1):74–88, 2012.

[11] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[12] A. Copeland. A "reasonable" social welfare function. Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences, University of Michigan, 1951.

[13] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012.

[14] E. Elkind and G. Erdélyi. Manipulation under voting rule uncertainty. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 627–634. IFAAMAS, June 2012.

[15] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pages 299–310. Springer-Verlag *Lecture Notes in Computer Science #5814*, October 2009.

[16] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 768–773. AAAI Press, July 2010.

[17] G. Erdélyi, L. Piras, and J. Rothe. The complexity of voter partition in Bucklin and fallback voting: Solving three open problems. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 837–844. IFAAMAS, May 2011.

[18] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[19] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[20] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[21] P. Faliszewski and A. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[22] L. Hačijan. A polynomial algorithm in linear programming. *Soviet Math. Dokl.*, 20:191–194, 1979.

[23] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[24] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[25] J. Lang, M. Pini, F. Rossi, D. Salvagnin, K. Venable, and T. Walsh. Winner determination in voting trees with incomplete preferences and weighted votes. *Journal of Autonomous Agents and Multi-Agent Systems*, pages 130–157, 2012.

[26] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[27] M. Pini, F. Rossi, K. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation: Complexity results. *Artificial Intelligence*, 175(7):1272–1289, 2011.

[28] L. Xia and V. Conitzer. Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011.

[29] L. Xia, J. Lang, and J. Monnot. Possible winners when new alternatives join: New results coming up! In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 829–836. IFAAMAS, May 2011.

Dorothea Baumeister, Magnus Roos, Jörg Rothe, and Lena Schend
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: `{baumeister,roos,rothe,schend}@cs.uni-duesseldorf.de`

Lirong Xia
Center for Research on Computation and Society (CRCS)
Harvard University
Cambridge, MA 02138, USA
Email: `lxia@seas.harvard.edu`

College admissions with stable score-limits

Péter Biró[1] and Sofya Kiselgof[2]

Abstract

A common feature of the Hungarian, Irish, Spanish and Turkish higher education admission systems is that the students apply for programmes and they are ranked according to their scores. Students who apply for a programme with the same score are in a tie. Ties are broken by lottery in Ireland, by objective factors in Turkey (such as date of birth) and other precisely defined rules in Spain. In Hungary, however, an equal treatment policy is used, students applying for a programme with the same score are all accepted or rejected together. In such a situation there is only one question to decide, whether or not to admit the last group of applicants with the same score who are at the boundary of the quota. Both concepts can be described in terms of *stable score-limits*. The strict rejection of the last group with whom a quota would be violated corresponds to the concept of H-stable (i.e. higher-stable) score-limits that is currently used in Hungary. We call the other solutions based on the less strict admission policy as L-stable (i.e. lower-stable) score-limits. We show that the natural extensions of the Gale-Shapley algorithms produce stable score-limits, moreover, the applicant-oriented versions result in the lowest score-limits (thus optimal for students) and the college-oriented versions result in the highest score-limits with regard to each concept. When comparing the applicant-optimal H-stable and L-stable score-limits we prove that the former limits are always higher for every college. Furthermore, these two solutions provide upper and lower bounds for any solution arising from a tie-breaking strategy. Finally we show that both the H-stable and the L-stable applicant-proposing score-limit algorithms are manipulable.

Keywords: stable matchings, college admissions, ties, manipulation
JEL classification: C78, I21

1 Introduction

Gale and Shapley [12] introduced a model and solution concept to solve the college admissions problem fifty years ago. In their model they suppose that the students submit preference lists containing the colleges they apply to, and each college ranks their applicants in a strict order and also provides an upper quota. Based on the submitted preferences a central body computes a fair solution. The fairness criterion they proposed is *stability*, which essentially means that if an application is rejected then it must be the case that the college must have filled its quota with applicants better than the our applicant's concerned. They gave an efficient method to find a stable matching and they proved that is actually optimal for the students in that sense that no student can be admitted to a better college in another stable matching. The Gale-Shapley algorithm has linear time implementation (see e.g. Knuth), which means that the running time of the algorithm is proportional to the number of applications. Another attractive property of this matching mechanism, proved by Roth , that it is strategy-proof for the students, i.e., no student can be admitted to any better college by submitting false preferences.

Later, it turned out (Roth [16]) that the algorithm proposed by Gale and Shapley had already been implemented in 1952 in the National Resident Matching Program and has

[1]This work was supported by OTKA grant K69027 and by the Hungarian Academy of Sciences under its Momemtum Programme (LD-004/2010).

[2]This work is partially supported by DecAN Laboratory NRU HSE.

been used since to coordinate junior doctor recruitment in the US. Moreover, the very same method has been implemented recently in the Boston [4] and New York [3] high school matching programs. However, college admissions are still organized in a completely decentralized way in the US, with all its flaws, that is unraveling through early admissions and the coordination problems caused by too many or not enough students admitted. See some representative stories on American college admissions practices in the blog of Al Roth [26].

There are many other countries where higher education admissions are more regulated, but yet not centralized. In Russia, the common timetable of the admissions prevent the unraveling and the use of 'original documents' provide better coordination regarding the number of students admitted, but yet the solution is far from being optimal.[3] In the UK, there is a common platform to manage the admissions by UCAS [27] but there is no centralized matching mechanism, the decisions and actions of the users (students and higher education institutions) are still decentralized.

Finally, there are some countries which do have centralized matching schemes for higher education admissions. In particular, there are scientific papers on the Chinese [19, 20], German [9, 18, 22], Hungarian [6, 7], Spanish [15], Turkish [5] schemes.[4]

The Chinese higher education admissions system is certainly the largest in the world, with more that 20 million students enrolled in 2009 [20]. The system is based on a centralized exam, called National College Entrance Examinations, which provides a score assigned to each students and this induce a ranking of the students by universities. The matching process (see [19]) is a kind of Boston-mechanism with some extra tweaks that makes the system manipulable and controversial. The German clearinghouse for higher education admissions deals only with a small segment of subjects (about 13,000 student from the total 500,000, see [22]). The clearinghouse is a mixed system, in the first phase the Boston-mechanism is used and in the second phase the college-proposing Gale-Shapley, so the process is not incentive compatible [9, 18].

The Hungarian, Irish, Spanish and Turkish higher education matching schemes are all based on a centralized scoring system. The Irish system has not been described yet in a scientific paper to the best of our knowledge.[5] In the other three countries students are assigned a score with regard to each programme they applied to, these scores are coming mainly from their grades and entrance exams. The scores of a student may differ at two programmes, since when calculating the score of a student for a particular programme only those subjects are considered which are relevant for that programme. The solution of the admission processes are represented by the so-called *score-limits*, which are referred to as 'base scores' in Turkey [5] and 'cutoff marks' in Spain. The score-limit of a programme means the lowest score that allows a student to be admitted to that programme. The score-

[3]Each applicant applies to at most five universities, but does not inform universities about her preferences among them. Universities rank students using results of Unified State Exams. Two 'admission rounds' are organized that are similar to the first two steps of a deferred acceptance procedure. After the second step, universities that still have empty seats are allowed to organize additional admissions.

[4]However, we shall note that regrettably these scientific papers deal only with some special features of these systems (as we also do in this paper) so not all the aspects of these schemes are described. But luckily, there is a new European research network, called Matching in Practice [25], one of whose aim is to collect and describe current matching practices in Europe. So hopefully we will have a better picture and understanding on the current practices, at least in Europe.

[5]From the information published at the website of the Central Applications Office [21] it seems that the college-proposing Gale-Shapley algorithm is used in Ireland with some special features. One is that students can apply for 'level 8' and 'level 7/6' courses simultaneously, and these applications are processed separately, so a student may receive more than one offer at a time. There are deadlines for accepting offers and if offers are rejected then further offers are made by the higher education institutions, so the mechanism is somewhat decentralized. The tie-breaking is based on 'random-numbers' assigned to students with regard to each programme they applied for, so the ties are broken differently for different programmes involving perhaps the same applicants.

limits together with the preferences of the students naturally induce a matching, where each student is admitted to the first place on her list where she achieved the score-limit.

In Turkey [5] the ties are broken according to the date of birth of the students and the college-proposing Gale-Shapley algorithm is used. In Spain the scoring method is fine enough (the admission marks are from 5 to 14 with 3 decimal fractions, and some further priority rules are also used), so ties are very unlikely. They use the applicant-proposing Gale-Shapley algorithm with the special feature of limiting the length of the preference lists, a setting that creates strategic issues that were studied in detail by Romero-Medina [15] and Calsamiglia et al. [10].

In fact, in most applications where ties may occur, the programme coordinators break these ties. In the high school matching schemes in New York [3] and Boston [4] lottery is used for breaking ties. However, this may lead to suboptimal solutions as Erdil and Erkin [11] pointed out, but according to the study by Abdulkadiroglu et al [1] this is the only way to keep the mechanism strategy-proof. In the Scottish Foundation Allocation Scheme [24], where the junior doctors are matched to hospitals, the organizers attempt to break the ties in such a way that in the resulted matching as many doctors are allocated as possible (see Irving and Manlove [14]).[6]

In contrast, in the Hungarian higher education admission scheme [23] the ties are not broken, therefore the students applying for a particular programme with equal scores are either all accepted or all rejected. We call this an *equal treatment* policy.

In particular, the ties are handled in the following way in Hungary. No quota may be violated, so the last group of students with the same score, with whom the quota would be exceeded, are all rejected. There is however an alternative policy that could be followed where the quotas may be exceeded by the admission of the last group of students with the same score, but only if there were unfilled places left otherwise.

As we will show in Section 3, both concepts can lead to matchings that satisfy special stability conditions based on score-limits that we formalize in Section 2. We refer to the first, more restrictive solution as *H-stable* (i.e., higher-stable) *score-limits* and we call the second, more permissive solution *L-stable* (i.e., lower-stable) *score-limits*. Note that these stable score-limit concepts generalize the original notion of stability by Gale and Shapley, since they are equivalent to that if no tie occurs. In Section 4, we show how one can extend the Gale-Shapley algorithm to find H-stable and L-stable score-limits. Moreover, in Section 5 we prove that the applicant-oriented versions provide the minimal stable score-limits (therefore they are the best possible solutions for the applicants), whilst the college-oriented versions provide maximal stable score-limits (therefore, they are the worst possible solutions for the applicants).

Furthermore, we show in Section 5 that comparing the H-stable and L-stable score-limits, the L-stable score-limits are more favorable for the applicants as they are lower. In particular, we show that no college can have a higher score-limit in the applicant-optimal L-stable solution than in the applicant-optimal H-stable solution (and the same applies for the applicant-pessimal solutions produced by the college-oriented versions). Interestingly, we also show that the applicant-optimal solution produced after a tie-breaking is always between these two kinds of solutions. Therefore the matchings corresponding to the H-stable and L-stable score-limits may provide upper and lower bounds for every applicant regarding her match in a scheme which uses any kind of tie-breaking strategy. Finally, in Section 6 we give examples showing that neither the H-stable nor the L-stable version of

[6]In SFAS [24], applicants are ranked by NHS Education for Scotland in a so-called master list, in order of score each applicant has a numerical score allocated partly on the basis of academic performance and partly as a result of the assessment of their application form. The range of possible scores (approximately 40 100) is much smaller than the number of applicants (around 750 each year), so there are ties of substantial length in the master list.

the applicant-oriented score-limit algorithm is strategy-proof. We conclude in Section 7.

2 The definition of stable score-limits

Let $A = \{a_1, a_2, \ldots, a_n\}$ be the set of applicants and $C = \{c_1, c_2, \ldots, c_m\}$ be the set of colleges, where q_u denotes the quota of college c_u. Let the ranking of the applicant a_i be given by a preference list P^i, where $c_v >_i c_u$ denotes that c_v precedes c_u in the list, i.e. the applicant a_i prefers c_v to c_u. Let s_u^i be a_i's final score at college c_u. Final scores are positive numbers, as in practice the students with scores below a common minimum threshold are rejected automatically (this minimum score is currently 200 in Hungary, and it applies for every study).

The *score-limits* of the colleges are represented with a non-negative integer mapping $l : C \to \mathbb{N}$. An applicant a_i is admitted to a college c_u if she achieves the score-limit at college c_u, and that is the first such place in her list, i.e. when $s_u^i \geq l(c_u)$, and $s_v^i < l(c_v)$ for every college c_v such that $c_v >_i c_u$.

If the score-limits l imply that applicant a_i is allocated to college c_u, then we set the Boolean variable $x_u^i(l) = 1$, and 0 otherwise. Let $x_u(l) = \sum_i x_u^i(l)$ be the number of applicants allocated to c_u under score-limits l.

Furthermore, let $l^{u,t}$ be defined as follows: $l^{u,t}(c_u) = l(c_u) + t$ and $l^{u,t}(c_v) = l(c_v)$ for every $v \neq u$. That is, we increase the score-limit of college c_u by t (or decrease it if t is negative), but we leave the other score-limits unchanged.

To introduce the H-stable and L-stable score-limits, first we define the corresponding feasibility notions. Score-limits l are *H-feasible* if $x_u(l) \leq q_u$ for every college $c_u \in C$. That is, the number of applicants may not exceed the quota at any college. This means that the last group of students with equal scores, with whom the quota would be exceeded, are all rejected. Score-limits l are *L-feasible* if for every college $c_u \in C$ such that $x_u(l) \geq q_u$ it must be the case that $x_u(l^{u,1}) < q$. So the quotas may be exceeded at any college, but only with the worst group of students who are admitted there with equal scores.

We say that score-limits l are *H-stable* (resp. *L-stable*) if l are H-feasible (L-feasible) and for each college c_u either $l(c_u) = 0$ or $l^{u,-1}$ are not H-feasible (resp. L-feasible). Thus H-stability means that we cannot decrease the score-limit of any college without violating its quota assuming that the others do not change their limits. L-stability means that no college c_u can admit a student if at least q_u of its current assignees have a higher score, but otherwise the score limits must be as small as possible. H-stability is the concept that is currently applied in the Hungarian higher education matching scheme.

We note that if no tie occurs (i.e. every pair of applicants have different scores at each college), then the two feasibility and stability conditions are the same and they are both equivalent to the original stability concept defined by Gale and Shapley. The correspondence between stable score-limits and stable matchings in case of strict preferences was first observed by Balinski and Sönmez [5] in relation with the Turkish college admissions scheme (where ties do not occur due to a tie-breaking strategy based on the age of the applicants). Furthermore Azevedo and Leshno [2] have also used this observation in a general college admissions model involving continuum number of students.

3 Stable score-limit algorithms

Both the H-stable and L-stable score-limit algorithms are natural extensions of the Gale–Shapley algorithm. The only difference is that now, the colleges cannot necessarily select exactly as many best applicants as their quotas allow, since the applicants may have equal scores. If the scores of the applicants are all different at each college then these algorithms are equivalent to the original one. In this section we will present the applicant-proposing and the college-proposing score-limit algorithms. For simplicity we describe these algorithms with regard to the H-stability concepts only and we add some information about the L-stable versions in brackets whenever they differ from the H-stable versions.

College-oriented algorithms:

In the first stage of the algorithm, let us set the score-limit at each college independently to be the smallest value such that, when all applicants are considered, the number of applicants offered places does not exceed its quota (resp. may exceed the quota but only if without the last tie of these students the quota is unfilled). Let us denote these score-limits by l_1. Obviously, there can be some applicants who are offered places by several colleges. These applicants keep their best offer, and reject all the less preferred ones, moreover they also cancel their less preferred applications.

In the subsequent stages, the colleges check whether their score-limits can be further decreased, since some of their offers may have been rejected in the previous stage, hence they look for new students to fill the empty places. So each college sets its score-limit independently to be the least possible that keeps the solution H-feasible (resp. L-feasible) considering their actual applications. If an applicant get a proposal from some new, better college, then she accepts the best offer, at least temporarily, and rejects or cancels her other, less preferred applications.

Formally, let l_k be the score-limit after the k-th stage. In the subsequent stage, at each college c_u, the largest integer t_u is chosen, such that $t_u \leq l_k(c_u)$ and $x_u(l_k^{u,-t_u}) \leq q_u$ (resp. if $x_u(l_k^{u,-t_u}) \geq q_u$ then $x_u(l_k^{u,-t_u+1}) < q_u$). That is, by decreasing its score-limit by the largest score t_u that keeps the solution H-feasible, i.e., where the number of applicants offered a place by c_u does not exceed its quota (resp. may exceed the quota but only if without the last tie of these students the quota is unfilled), by supposing that all other score-limits remained the same. For each college c_u let $l_{k+1}(c_u) := l_k^{u,-t_u}(c_u)$ be the new score-limit. Again, some applicants can be offered a place by more than one college, so $x_u(l_{k+1}) \leq x_u(l_k^{u,-t_u})$. Obviously, the new score-limits remain feasible.

Finally, if no college can decrease its score-limit then the algorithm stops. The H-stability (resp. L-stability) of the final score-limits is obvious by definition. Let us denote the corresponding solutions of the H-stable and L-stable versions by l_C^H and l_C^L, respectively.

Applicant-oriented algorithms:

Let each applicant propose to her first choice in her list. If a college receives more applications than its quota, then let its score-limit be the smallest value such that the number of provisionally accepted applicants does not exceed its quota (resp. may exceed the quota but only if without the last tie of these students the quota is unfilled). We set the other score-limits to be 0.

Let the score-limits after the k-th stage be l_k. If an applicant has been rejected in the k-th stage, then let her apply to the subsequent college in her list, say c_u, where she achieves the actual score-limit $l_k(c_u)$, if there remains such a college in her list. Some colleges may receive new proposals, so if the number of provisionally accepted applicants exceeds the quota at a college (resp. exceeds the quota and without the last tie of these students the quota is still filled), then it sets a new, higher score-limit $l_{k+1}(c_u)$.

Again, for each such college c_u, this is the smallest score-limit such that the number of applicants offered a place by c_u does not exceed its quota (resp. may exceed the quota but only if without the last tie of these students the quota is unfilled), by supposing that all other score-limits remained the same. This means that c_u rejects all those applicants that do not achieve this new limit.

The algorithm stops if there is no new application. The final score-limits are obviously H-feasible (resp. L-feasible). The solution is also H-stable (resp. L-stable), because after a score-limit has increased for the last time at a college, the rejected applicants get less preferred offers during the algorithm. So if the score-limit in the final solution were decreased by one for this college, then these applicants would accept the offer, and the solution would not remain H-feasible (resp. L-feasible). Let us denote the corresponding solutions by the H-stable and L-stable applicant-oriented versions by l_A^H and l_A^L, respectively. The following result is therefore immediate.

Theorem 3.1. *The score-limits l_C^H and l_C^L obtained by the college-oriented score-limit algorithms are H-stable and L-stable, respectively. The score-limits l_A^H and l_A^L obtained by the applicant-oriented score-limit algorithms are H-stable and L-stable, respectively.*

4 Optimality of the outputs

It is easy to give an example to show that not only some applicants can be admitted by preferred places in l_A^H as compared to l_C^H, but the number of admitted applicants can also be larger in l_A^H (and the same applies for the L-stable setting). We say that score-limits l are *better* than l_* for the applicants if $l \leq l_*$, i.e., if $l(c_u) \leq l_*(c_u)$ for every college c_u. In this case every applicant is admitted to the same or to a preferred college under score-limits l than under l_*.

Theorem 4.1. *Given a college admission problem with scores, l_C^H are the worst possible and l_A^H are the best possible stable score-limits for the applicants, i.e. for any H-stable score-limits l, $l_A^H \leq l \leq l_C^H$ holds.*

Proof. Suppose first for a contradiction that there exists a H-stable score-limit l_* and a college c_u such that $l_*(c_u) > l_C^H(c_u)$. During the college-oriented algorithm there must be two consecutive stages with score-limits l_k and l_{k+1}, such that $l_* \leq l_k$ and $l_*(c_u) > l_{k+1}(c_u)$ for some college c_u.

Obviously, $l_k^{u,-t_u}(c_u) = l_{k+1}(c_u)$ by definition. Also, $x_u(l_k^{u,-t_u}) \leq q_u < x_u(l_*^{u,-1})$, where the first inequality holds by definition of t_u, as we choose the new limit for college c_u such a way that the number of temporarily admitted applicants does not exceed its quota. The second inequality holds by the H-stability of l_*. So there must be an applicant, say a_1, who is admitted to c_u at $l_*^{u,-1}$ but not admitted to c_u at $l_k^{u,-t_u}$.

On the other hand, the indirect assumption implies that $l_k^{u,-t_u}(c_u) = l_{k+1}(c_u) \leq l_*(c_u) - 1 = l_*^{u,-1}(c_u)$. Applicant a_1 has a score of at least $l_k^{u,-t_u}(c_u)$, which is enough to be accepted to c_u, so she must be admitted to some college c_v under $l_k^{u,-t_u}(c_u)$ which is preferred to c_u. Obviously a_1 must be also admitted to c_v under l_k. But the H-stability of l_* implies that $l_*(c_v) > l_k(c_v)$, a contradiction.

To prove the other direction, we suppose for a contradiction that there exists H-stable score-limits l_* and a college c_u such that $l_*(c_u) < l_A^H(c_u)$. During the applicant-oriented algorithm there must be two consecutive stages with score-limits l_k and l_{k+1}, such that $l_* \geq l_k$ and $l_*(c_u) < l_{k+1}(c_u)$ for some college c_u. At this moment, the reason for the incrementation is that more than q_u students are applying for c_u with a score of at least $l_*(c_u)$. This implies that one of these students, say a_i, is not admitted to c_u under l_* (however she has a score of at least $l_*(c_u)$ there). So, by the H-stability of l_*, she must be admitted to a preferred college, say c_v under l_*. Consequently, a_i must have been rejected by c_v in a previous stage of the algorithm, and that is possible only if $l_*(c_v) < l_k(c_v)$, a contradiction. □

Theorem 4.2. *Given a college admission problem with scores, l_C^L are the worst possible and l_A^L are the best possible L-stable score-limits for the applicants, i.e. for any L-stable score-limits l, $l_A^L \leq l \leq l_C^L$ holds.*

Proof. Suppose first for a contradiction that there exist stable score-limits l_* and a college c_u such that $l_*(c_u) > l_C^L(c_u)$. During the college-oriented algorithm there must be two consecutive stages with score-limits l_k and l_{k+1}, such that $l_* \leq l_k$ and $l_*(c_u) > l_{k+1}(c_u)$ for some college c_u.

This assumptions imply that $x_u(l_k^{u,-t_u+1}) < q_u \leq x_u(l_*)$. Here, the first inequality holds by the L-feasibility of l_{k+1}, and the second inequality by the L-stability of l_*. At the same time, by our assumption, $l_*(c_u) > l_{k+1}(c_u)$, so $l_*(c_u) \geq l_{k+1}(c_u) + 1 = l_k^{u,-t_u+1}(c_u)$.

From the two above statements it follows that there must be an applicant, say a_1, who has a score $s_u(a_1) \geq l_*(c_u)$ and is admitted to c_u under l_*, but is not admitted to c_u under $l_k^{u,-t_u+1}$. So a_1 must have a seat at some college c_v under $l_k^{u,-t_u+1}$ such that $c_v >_{a_1} c_u$. Obviously, a_1 is also admitted to c_v under l_k. But a_1 is not admitted to c_v under l_*, therefore $l_k(c_v) < l_*(c_v)$, a contradiction.

To prove the other direction, we suppose for a contradiction that there exist stable score-limits l_* and a college c_u such that $l_*(c_u) < l_A^L(c_u)$. During the applicant-oriented algorithm there must be two consecutive stages with score-limits l_k and l_{k+1}, such that $l_* \geq l_k$ and $l_*(c_u) < l_{k+1}(c_u)$ for some college c_u.

At this moment, the reason for the incrementation is that more than q_u students are applying for c_u with score at least l_*, and c_u can choose a new score-limit $l_{k+1}(c_u) = l_k^{u,-t_u}(c_u)$, where $t_u > l_*(c_u) - l_k(c_u)$.

This implies that one of those students, who are admitted by c_u under l_{k+1}, say a_1, is not admitted to c_u under l_*. However she has a score higher than score-limit $l_*(c_u)$ there. So, by the L-stability of l_*, she must be admitted to a preferred college, say c_v, under l_*. Consequently, in the applicant-proposing procedure a_1 must have been rejected by c_v at some previous stage, and that is possible only if $l_*(c_v) < l_k(c_v)$, a contradiction. □

5 Comparison of the H-stable and L-stable versions

Intuitively it seems that the L-stable version of the algorithm is more *applicant-friendly* than the H-stable version. It turns out that we can prove the following result.

Theorem 5.1. *The score-limits obtained in the L-stable version of the applicants-oriented procedure are always equal or lower than the score-limits obtained in the H-stable version of the applicant-oriented procedure: i.e. $l_A^L \leq l_A^H$.*

Proof. Part I. Some colleges may have number of admitted students less than or equal to their quota under l_A^H, i.e. $q_u - x_u(l_A^H) \geq 0$. Each college c_u has a "waiting" list of applicants, who would prefer to be admitted to c_u rather than to their currently assigned colleges.

Let us apply some random tie-breaking to the original preference relation of the colleges. Each applicant a_i will get a new score $p_u^i \geq s_u^i$ such that no two applicants will have the same score at any college. Moreover, the new scores satisfy the following condition: if $s_u^j < s_u^i$, then $p_u^j < s_u^i$. These p_u^i scores are positive real numbers. For example, if there are three applicants with scores $s_u^1 = s_u^2 = 1$, $s_u^3 = 2$, the new scores might be $p_u^1 = 1$, $p_u^2 = 1.5$, $p_u^3 = 2$.

After that the following procedure is organized. If the number of applicants on c_u college's waiting list is more than the number of empty seats then college c_u sets it's new score-limit $m_A^H(c_u) \leq l_A^H(c_u)$ equal to the score p_u^i of the last admitted applicant in its waiting list. Otherwise let $m_A^H(c_u) = 0$. Note that the new score-limits m_A^H are non-negative real numbers. This means that each college make offers to applicants from its waiting list who fit the new score-limit.

Some applicants may receive more than one proposal. Each applicant accepts one, from the most preferred college, and rejects the others. If there remain any empty seat in colleges then the second step is organized in the same manner and so on. Thus essentially we run a college-proposing deferred-acceptance procedure with regard to the new scores. At the end of this procedure some new score-limits m^R are achieved such that $m^R \leq l_A^H$ by construction. These new score-limits m^R and the corresponding matching μ^R are stable (in the Gale-Shapley sense) according to new strict preferences of colleges, also by construction.

Part II. For the strict preference profile and corresponding scores p_u^i from Part I we can organize applicant-proposing deferred acceptance procedure (which is, in case of strict preferences, equivalent to both the H-stable and L-stable applicant-oriented algorithms). The resulting matching μ_A^R is, of course, stable under strict preferences. Furthermore, we can define score-limits m_A^R that are equal to the score of the last accepted applicant if college has no empty seats and to 0 otherwise. These score-limits m_A^R must be the lowest among all stable score-limits by the optimality theorem of Gale and Shapley. Therefore $m_A^R \leq m^R$ in particular.

Part III. Now we deal with m_A^R score-limits. Let us get back to the original weak order preferences of the colleges and corresponding applicants' scores s_u^i. For each college with $x_u(l_A^R) = q_u$ we can construct a "waiting" list of applicants, who prefer college c_u to their current matches under m_A^R.

Let us now apply the L-feasibility concept. At the first stage each college sets it's new score-limit $l_A^R(c_u) \leq m_A^R(c_u)$, that is the largest value, which allows to admit equal or more than the quota under weak order preferences as L-feasibility prescribes. For example, if there are two applicants with the same score s_u^i, such that one of them is admitted to c_u under m_A^R and the other is on the waiting list then we have to 'treat them equally', so we should lower the score-limit. Each college makes offers to these additional applicants.

Some applicants may receive more than one offer from colleges; in this case each applicant chooses the most preferred college. After that if there is any college with number of admitted applicants less than its quota then a new round starts. Each college chooses new, lower, L-feasible limit, and so on. That is we run the college-proposing score-limit procedure under L-stability. At the end, some new score-limits l^L are achieved such that $l^L \leq m_A^R$ by construction. These new score limits are L-feasible and L-stable, obviously.

Part IV. For each L-stable score-limit l^L we know that $l_A^L \leq l^L$ from Theorem 4.2, where l_A^L are stable score-limits obtained by the L-stable applicant-oriented algorithm.

Now we can construct the following inequalities: $l_A^L \leq l^L \leq m_A^R \leq m^R \leq l_A^H$. So we can conclude that for any college admissions problem with score-limits the outcome by the L-stable applicant-oriented algorithm is better for the applicants (i.e. yields lower score-limits) than the outcome of the H-stable applicant-oriented algorithm.

□

Theorem 5.2. *The score-limits obtained in the L-stable version of the college-oriented procedure are always equal or lower than the score-limits obtained in the H-stable version of the college-oriented procedure: i.e.* $l_C^L \leq l_C^H$.

Proof. Part I. Let us consider the l_C^L score-limits. Some colleges may have number of admitted students more than or equal to their quota, $x_u(l_C^H) \geq q_u$.

Let us apply a random tie-breaking to the original preference relation of the colleges. Each applicant a_i gets a new score $p_u^i \geq s_u^i$ such that no two applicants have the same score at any college, and these new scores do not contradict with the original ordering. Moreover, if $s_u^j < s_u^i$, then $p_u^j < s_u^i$). These p_u^i scores are positive real numbers.

After that the following procedure is organized. At the first stage each college sets its new score-limit $m_C^L(c_u) \geq l_C^L(c_u)$ such that according to the new scores p_u^i the number of applicants who fit this score-limit would be exactly q_u. The new score-limits m_C^L are non-negative real numbers. Let $m_C^L(c_u)$ be equal to 0 if the number of students admitted to c_u is less than c_u's quota and otherwise let $m_C^L(c_u)$ be equal to the lowest score p_u^i of any admitted student.

Some applicants are rejected from colleges they were assigned under l_C^L. Each rejected applicant then applies to the subsequent college in her list. Colleges receive new applications and, if necessary, raise the limits so that number of accepted applicants are equal to their quota. Some new applicants may be rejected, so a second round is organized in the same manner and so on. Thus we run an applicant-proposing deferred-acceptance procedure with respect to the perturbed strict preferences. At the end, some new score-limits m^R are obtained such that $m^R \geq l_C^L$ by construction. These new score-limits are stable (in the Gale-Shapley sense) according to the new strict preferences of colleges by construction.

Part II. For strict preference profile and corresponding scores p_u^i from Part I we can organize a college-oriented deferred-acceptance procedure. The resulting score-limits m_C^R are, of course, stable according to these strict preferences. Furthermore, the corresponding score-limits must be the lowest among all stable score-limits [12]. So, $m_C^R \geq m^R$.

Part III. Now we deal with m_C^R score-limits. For each college c_u, $x_u(l_A^R) \leq q_u$ holds under m_C^R. Each college c_u with number of assigned students lower than its quota has score-limit $l_A^R(c_u) = 0$. Now we get back to the original weak order preferences of the colleges and original applicants' scores s_u^i.

Let us now apply the H-feasibility concept. For each college we can construct a list of applicants, who prefer college c_u to their current matches under m_C^R. After that the following deferred acceptance procedure is organized. At the first stage each college sets it's new score-limit $l_C^R(c_u) \geq m_C^R(c_u)$ that is the smallest value, which allows to admit equal or less than the quota under weak order preferences as H-feasibility prescribes. Therefore some colleges may reject applicants. Each rejected applicant applies to the next college in her list. Colleges receive new applications and, if necessary, raise their score-limits in such a way that the number of accepted applicants is less than or equal to their quota. Some applicants may be rejected and a second round is organized in the same manner and so on. Thus we run an applicant-proposing deferred-acceptance procedure with regard to H-stability. At the end, each applicant is either accepted to some college or rejected by all acceptable colleges. Some new score-limits l^H are achieved such that $l^H \geq m_C^R$ by construction. These new score-limits are H-feasible and H-stable, obviously.

Part IV. For each H-stable score-limit l^H we know that $l_C^H \geq l^H$ from theorem 4.1, where l_C^H is a H-stable score-limit obtained by the applicant-oriented score-limit algorithm.

Now we can construct the following inequalities: $l_C^L \leq m^R \leq m_C^R \leq l^H \leq l_C^H$. So we can conclude that for any college admissions problem with score-limits the outcome by the L-stable college-oriented algorithm is better for the applicants (i.e. yields lower score-limits) than the outcome of the H-stable college-oriented algorithm.

□

Corollary 1. *Applicant-optimal H-stable and L-stable scorelimits (l_A^H and l_A^L) are upper and lower bounds (respectively) for scorelimits under any Pareto-optimal stable matching with random tie-breaking.*

6 Strategic issues

Here we give two examples showing that neither of the above described score-limit algorithms is strategy-proof. The manipulability from the applicants' side is only interesting in the case of applicant-oriented algorithms, as the applicants may successfully manipulate the college-oriented versions even for strict preferences (i.e., for scores with no ties). Therefore we only consider the applicant-oriented versions in the examples below.

Example 1. *Suppose that we have two colleges, c_u and c_v with one seat in each of them, and two applicants s_1 and s_2 applying to both c_u and c_v with a preference towards c_u and with equal scores at both places. So the preference list of the colleges and students are as follows.*

$$a_1 : c_u, c_v \qquad c_u : (a_1, a_2)$$
$$a_2 : c_u, c_v \qquad c_v : (a_1, a_2)$$

Figure 1: An example for the manipulability of the H-stable applicant-proposing algorithm

Here the only stable solution is the empty matching (i.e., score-limits higher than the scores of a_1 and a_2 at both colleges). However, if either of the students, say a_1 withdraws her application at c_u then the unique H-stable solution (under falsified preferences) is matching where a_1 is allocated to c_v and a_2 is allocated to c_u. So the manipulator (and actually the other student also) would improve.

The following example is essentially the same as the one that Hatfield and Milgrom [13] constructed in a different setting but for a similar purpose.

Example 2. *Suppose that we have two colleges, c_u and c_v with one seat in each of them, and three applicants a_1, a_2 and a_3 applying to both c_u and c_v with the following scores, $s_u^1 = 1$, $s_u^2 = 1$, $s_u^3 = 2$, $s_v^1 = 3$, $s_v^2 = 2$ and $s_v^3 = 1$. These can be described equivalently with the preference lists below.*

$$a_1 : c_u, c_v \qquad c_u : a_3, (a_1, a_2)$$
$$a_2 : c_v, c_u \qquad c_v : a_1, a_2, a_3$$
$$a_3 : c_v, c_u$$

Figure 2: An example for the manipulability of the L-stable applicant-proposing algorithm

Here the only L-stable solution is the matching $\{(a_1, c_v), (a_3, c_u)\}$ (i.e., with score-limits $l(c_u) = 2$ and $l(c_v) = 3$). However, if a_2 were to reverse her preferences with regard to the two colleges then the L-stable applicant-oriented algorithm would produce the matching $\{(a_1, c_u), (a_2, c_u), (a_3, c_v)\}$, where the manipulator (and actually both the two other applicants) would improve.

7 Further notes

In this paper we studied the concept of stable score-limits for higher education admissions. In particular we introduced and analyzed the notions of H-stability and L-stability when ties occur, a situation currently present in the Hungarian scheme.

As future research, we would like to investigate the college admissions practices of other countries, in particular those which have centralized systems based on score-limits. Regarding the Hungarian application, we would like to conduct an experiment with real data and compute the four possible extreme stable score-limits, namely the applicant-optimal vs applicant-pessimal score-limits under H-stability and L-stability. Finally, it would be also interesting to see how these concepts can be used in other settings, e.g. what could be the corresponding solutions for the Boston and New York high school matching programs.

Regarding the theoretical problems, we would like to investigate whether there is any structure behind the H-stable and L-stable score-limits. It would be also worth to study further the relation of solutions satisfying equal treatment policy and those produced by tie-breaking strategies. For instance, one may can show some intuitive statements such that finer scoring methods lead to solutions 'closer' to the stable matchings obtained by tie-breaking strategies, and finer scoring methods are 'harder' to manipulate.

Acknowledgment

We would like to thank to Gábor Varjasy, the representative of Higher Education National Institute. Further we acknowledge Tamás Fleiner and Rob Irving for their useful comments and we also thank Jordi Masso and Antonio Romero-Medina for their help in understanding the Spanish higher education admissions system.

References

[1] A. Abdulkadiroğlu, P.A. Pathak, and A.E. Roth. Strategy-proofness versus efficiency in matching with indifferences: Redesigning the NYC high school match. *American Economic Review*, 99(5):1954–1978, 2009.

[2] E. Azevedo and J. Leshno. A supply and demand framework for two-sided matching markets. *mimeo*, 2011.

[3] A. Abdulkadiroğlu, P.A. Pathak, and A.E. Roth. The New York City high school match. *American Economic Review, Papers and Proceedings*, 95(2):364–367, 2005.

[4] A. Abdulkadiroğlu, P.A. Pathak, A.E. Roth, and T.Sönmez. The Boston public school match. *American Economic Review, Papers and Proceedings*, 95(2):368–371, 2005.

[5] M. Balinski and T. Sönmez. A tale of two mechanisms: Student placement. *J. Econom. Theory*, 84(1):73–94, 1999.

[6] P. Biró. Student admissions in Hungary as Gale and Shapley envisaged. *Technical Report no. TR-2008-291 of the Computing Science Department of Glasgow University*, 2008.

[7] P. Biró, T. Fleiner, R.W. Irving, and D.F. Manlove. College admissions with lower and common quotas. *Theoretical Computer Science*, 411:3136–3153, 2010.

[8] P. Biró, F. Klijn. Matching with Couples: a Multidisciplinary Survey. To appear in *International Game Theory Review*, 2012.

[9] S. Braun, N. Dwenger, and D. Kübler. Telling the truth may not pay off: An empirical study of centralised university admission in Germany. 2010. The B.E. Journal of Economic Analysis and Policy, 10(1): Article 22.

[10] C. Calsamiglia, G. Haeringer, and F. Klijn. 'Constrained School Choice: An Experimental Study. *American Economic Review*, 100(4):1860–1874, 2010.

[11] A. Erdil and H. Erkin. What's the matter with tie-breaking? improving efficiency in school choice. *American Economic Review*, 98:669–689, 2008.

[12] D. Gale and L.S. Shapley. College Admissions and the Stability of Marriage. *Amer. Math. Monthly*, 69(1):9–15, 1962.

[13] J.W. Hatfield and P.R. Milgrom. Matching with contracts. *American Economic Review*, 95(4):913–935, 2005.

[14] R.W. Irving and D.F. Manlove. Finding large stable matchings. *ACM Journal of Experimental Algorithmics*, 14, section 1, article 2, 30 page.

[15] A. Romero-Medina. Implementation of stable solutions in a restricted matching market. *Review of Economic Design*, 3(2):137–147, 1998.

[16] A.E. Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of Political Economy*, 6(4):991–1016, 1984.

[17] C.-P. Teo, Y. Sethuraman, and W.-P. Tan. Gale-Shapley stable marriage problem revisited: Strategic issues and applications. *Management Science*, 47:1252–1267, 2001.

[18] A. Westkamp. An analysis of the German university admissions system. *mimeo, University of Bonn*, 2011.

[19] H. Zhang. An Analysis of The Chinese College Admission System. *PhD Thesis, University of Edinburgh*, 2009.

[20] Y. Zhang. The determinants of National College Entrance Exam performance in China - with an analysis of private tutoring. *PhD Thesis, Columbia University*, 2011.

[21] Central Applications Office, Ireland website. www.cao.ie

[22] D. Kübler. University admission practices - Germany. matching-in-practice.eu, accessed on 23 May 2012

[23] P. Biró. University admission practices - Hungary. matching-in-practice.eu, accessed on 23 May 2012

[24] R. Irving. Matching practices for entry-labor markets - Scotland. matching-in-practice.eu, accessed on 23 May 2012

[25] Matching in Practice website. matching-in-practice.eu

[26] Al Roth's market designer blog. http://marketdesigner.blogspot.com

[27] UCAS website. http://www.ucas.com

Péter Biró
Institute of Economics, Research Centre for Economic and Regional Studies
Hungarian Academy of Sciences
H-1112, Budaörsi út 45, Budapest, Hungary
Email: birop@econ.core.hu

Sofya Kiselgof
Laboratory of Decision Choice and Analysis (DecAn)
NRU Higher School of Economics, Moscow
Moscow, Pokrovskiy Boulevard, 11, Russia
Email: skiselgof@hse.ru

Robust Winners and Winner Determination Policies under Candidate Uncertainty

Craig Boutilier, Jérôme Lang, Joel Oren and Héctor Palacios

Abstract

We consider voting situations in which a group considers a set of options or candidates, but where some candidates may turn out to be unavailable. If determining availability is costly (e.g., in terms of money, time, or computation), it may be beneficial for the group to vote prior to determining candidate availability, and only test the winner's availability after the vote. However, since few voting rules are robust to candidate deletion, winner determination usually involves a number of such availability tests. We outline a model for analyzing such problems. We define *robust winners* relative to potential candidate unavailability, a notion that is tightly related to control by candidate addition. Then assuming a distribution over availability and costs for availability tests (or *queries*), we define an *optimal query policy* for a vote profile to be one with minimal expected query cost that determines the true winner. We describe a dynamic programming algorithm for computing optimal query policies, as well as a myopic heuristic approach using *information gain* to choose queries. Finally, we outline a number of theoretical and practical questions raised by our model.

1 Introduction

There are many social choice situations in which members of a group may need to specify their preferences over a set of alternatives or *candidates* without knowing whether any specific candidate is in fact viable or *available* for selection. Lu and Boutilier [12] propose the *unavailable candidate model* for studying situations in which potential winners are approached sequentially and the first available candidate is the winner (e.g., consider a hiring committee deciding on the order in which to make job offers, knowing that candidates may refuse). In this paper, we consider a more flexible situation in which determining candidate availability is *costly*, but does not commit one to selecting the first available candidate as the winner.

In our setting, voting over the set of *potential candidates* prior to determining availability can often make sense. For example, a group of friends deciding on a restaurant may attempt to (perhaps partially) determine their aggregate preferences prior to calling (or walking to) restaurants to find out if reservations are possible. A legislative body deciding amongst various public projects may vote prior to knowing their precise financial costs, since the process of budgeting—with engineering estimates, environmental impact studies, etc.—is itself costly. In an AI planning context, a group may vote on the collection of goals to pursue prior to knowing their feasibility, since determining the feasibility of any specific goal set involves solving a computationally difficult problem. In each of these cases, having some idea of who might potentially win an election can narrow the set of *availability tests* that need to be performed, hence the financial, time or computational cost of determining the true winner. But since almost no practical voting rules are robust to candidate deletion, reliably declaring a winner requires making *some* availability tests.

In this paper, we describe a model for addressing such problems, identify a number of key concepts and interesting computational questions that arise in this model, and make some first steps toward solving them. In rough outline, our model assume a set of *potential candidates* X, voter preferences over X in the form of a vote profile $\mathbf{v}$, and some voting rule r. Since candidate availability is uncertain, we assume some distribution P over subsets of X, where $P(A)$ is the probability of A being the true available set, and we assume that each candidate x can be *queried*, for a cost, to determine its availability. Our aim is to determine the winner $r(\mathbf{v}^{\downarrow A})$ of the election (where $\mathbf{v}^{\downarrow A}$

is the restriction of $\mathbf{v}$ to available candidates) in a way that minimizes expected query cost (e.g., number of phone calls, planning problems to be solved, etc.).

To this end, we are primarily interested in *query policies* that propose a (conditional) sequence, or tree, of queries designed to determine enough information about the available set A to declare a winner. Notice that we *need not know A precisely* to determine the winner of an election. Given responses to some set of queries, we know that some candidates $Q^+ \subseteq X$ are available, and some $Q^- \subseteq X$ are not. Relative to such an *information set* $\langle Q^+, Q^- \rangle$, we say x is a *robust winner* if $r(\mathbf{v}^{\downarrow A}) = x$ for any $Q^+ \subseteq A \subseteq X \setminus Q^-$. In other words, this information set is sufficient to determine the winner regardless of the availability of the remaining candidates (e.g., if x is a *majority* winner in a plurality election, the status of other candidates is irrelevant if x is available). The problem of determining a robust winner has very tight connections to the problem of control by candidate addition [2], as we discuss in Sec. 3.

Query policies need only ask enough queries to determine a robust winner. While computing robust winners can be computationally difficult for some voting rules, our primary concern is minimizing query costs, which are much more important than winner determination costs. In Sec. 4 we formulate the query problem as one of constructing a minimal cost *decision tree* over features corresponding to the availability of specific candidates, and whose goal is to classify available sets A according to their winners. We describe a (relatively) inexpensive dynamic programming algorithm for computing optimal query policies; but we also explore the use of standard decision tree induction methods based on *information gain* [16] which are much more computationally tractable. We also consider policies tuned to extreme availability probabilities. Finally, in Sec. 5 we outline a number of interesting theoretical questions, some of which whose answers can have further practical impact on the construction of optimal query policies.

2 Voting with Uncertain Candidate Availability

We first outline our model for winner determination with uncertainty in candidate availability and briefly discuss relevant related work.

2.1 The Model and Decision Problem

We assume a set of n voters N and a set of m *potential candidates* X, with each voter $i \in N$ having a complete, strict preference ordering or *vote* v_i over X, with *vote profile* $\mathbf{v}$ denoting the vector of all votes. A *voting rule* r maps every profile to a (unique) winning candidate (we assume ties are broken in some fashion). We consider rules such as plurality, Borda, and Copeland below, but our framework is completely general. Given a profile $\mathbf{v}$, let $m(\mathbf{v})$ be its majority graph, and $m(\mathbf{v})^*$ the transitive closure of $m(\mathbf{v})$. To make things simpler, we assume an odd number of voters so that $m(\mathbf{v})$ is a tournament. The *top cycle* $TC(\mathbf{v})$ of $\mathbf{v}$ is the set of all candidates x such that for all $y \neq x$, $(x, y) \in m(\mathbf{v})^*$. TC (plus a tie-breaking mechanism) is also considered as a voting rule. A voting rule r is *Smith-consistent* if $r(\mathbf{v}) \in TC(\mathbf{v})$ for any profile $\mathbf{v}$. We assume that r can be applied directly to profiles over arbitrary subsets of X: we use $\mathbf{v}^{\downarrow A}$ to denote the restriction of $\mathbf{v}$ to candidates A, obtained by deleting elements of $X \setminus A$ from each vote, and $r(\mathbf{v}^{\downarrow A})$ to denote the winner w.r.t. this restricted profile. We sometimes use the notation $w(A)$ to denote this winner, suppressing mention of r and $\mathbf{v}$.

We now turn our attention to the possibility that certain candidates in X may be unavailable. There are two natural ways to address this. First, we might first check the availability of all candidates in X, and elicit votes over the set of available candidates $A \subseteq X$. This has the advantage of minimizing vote elicitation costs: voters need not rank or compare unavailable candidates. However, if testing the availability of candidates is itself costly, as discussed above, this may make far more availability tests than needed given voter preferences.

A second approach is to first elicit voter rankings over the entire set X, then use this information to focus attention on "relevant" availability tests. This has the advantage of reducing the cost of availability tests. This is most appropriate when such tests are costly relative to preference assessment or elicitation, as in our examples above. Determining suitable tests is, nonetheless, far from straightforward. One obvious approach is to use the voting rule r to rank candidates, test availability in the order of this aggregate ranking, and select the first available candidate as the winner. However, this assumes that the choice function implemented by voting rule r is *rationalizable*, which is rarely the case. Consider the profile $\mathbf{v}$ with 4 votes abc, 3 votes bca and 2 votes cab. Ranking candidates by their plurality score gives aggregate ranking abc. The policy above, once learning a is available, would select a as the winner without further tests; but if we were to learn that b is unavailable and c available, the true plurality winner for $\mathbf{v}^{\downarrow\{a,c\}}$ is c. This paradox arises since, under mild conditions, no non-dictatorial voting rule is robust to the deletion of non-winning alternatives [4]. As a result, choosing a winner usually requires confirming the availability of specific candidates.[1] Furthermore, minimizing the costs of such availability tests is non-trivial.

We model candidate availability as follows: we partition X into a (possibly empty) *known set* $Y \subseteq X$ of candidates that are sure to be available, and an *unknown set* $U = X \setminus Y$ for which availability is uncertain.[2] Let $\mathcal{A}$ denote the family of subsets $Y \subseteq A \subseteq X$, where $A \in \mathcal{A}$ is a possible *available set*. The general unavailable candidate model [12] requires a distribution P over 2^U, where $P(S)$ denotes the probability that $S \subseteq U$ is the actual available set of candidates from those in U. We assume for simplicity that the availability of each candidate $x \in U$ is independent, given by probability p_x. This induces the obvious distribution over the available sets $\mathcal{A}$.

For any $x \in U$, we assume one can *query* x using some *availability test* to determine its availability (e.g., calling for a restaurant reservation, computing plan for some goal x), which incurs a cost c_x. Informally, a *query policy* (see Sec. 4) consists of a tree whose interior nodes are labeled by queries, edges by availability, and leaves by winners. We desire policies that, given a profile $\mathbf{v}$, accurately determine the winner w.r.t. the *actual* available set A with minimum expected query cost. After any sequence of queries and responses, we have refined information about the available set: an *information set* is an ordered pair $Q = \langle Q^+, Q^- \rangle$, where $Q^+ \subseteq U$, $Q^- \subseteq U$, and $Q^+ \cap Q^- = \emptyset$; intuitively, Q^+ (resp. Q^-) is the set of queries (candidates) for which positive (resp. negative) availability has been determined.

Clearly, winners can often be determined without full knowledge of candidate availability. In our example above, knowing that a and b are available suffices to declare a the winner: availability of c is irrelevant; knowing a is available and c is unavailable is also sufficient to select a.

Definition 1 *Let r be a voting rule, $\mathbf{v}$ a profile over candidate set X, and $Y \subseteq X$ a set of candidates known to be available. We say that $x \in Y$ is a* robust winner *w.r.t. $\langle X, Y, \mathbf{v}, r \rangle$ if, for any A such that $Y \subseteq A \subseteq X$, $r(\mathbf{v}(A)) = x$.*

Intuitively, a winner is robust if it not only wins in the original profile $\mathbf{v}$, but continues to win no matter which candidates in $X \setminus Y$ are deleted. In our setting, the existence of a robust winner relative to the current information set is necessary and sufficient to stop any querying process. Specifically, we say that information set Q is *r-sufficient* if there is a robust winner x w.r.t. $\langle X \setminus Q^-, Y \cup Q^+, \mathbf{v}, r \rangle$. For any r-sufficient information set Q, let $w(Q)$ denote this (unique) robust winner.

2.2 Related Work

Lu and Boutilier [12] study a setting where the set of available candidates is unknown at the time of voting, and assume a distribution over available sets A as we do. Unlike our model, they assume a's availability cannot be tested without "offering it the win," hence focus on computing optimal *ranking*

[1] At a minimum, one might require that the winner itself be available, but we consider exceptions to this below.

[2] If any candidates are known to be unavailable *a priori*, we remove them from set X.

policies, as discussed above (see also Baldiga and Green [1] who develop a related availability model with different motivations). Wojtas and Faliszewski [19] also consider candidates with uncertain availability in a counting version of *control by adding candidates* (see below), a problem that is closely related to ours. Their input consists of a subset of candidates Y known to be available, a distribution over subsets of $X \setminus Y$, and votes over X; given a fixed voting rule, they consider the complexity of computing the probability that a given candidate is the winner.

Considerable recent research, starting with Chevaleyre *et al.* [3], studies a variant of the possible winner problem where the candidate set is not known at the time of voting, but take the opposite perspective to ours. A *lower bound* on the candidate set is known initially, and new candidates may join after preferences for the initial set have been elicited. Efficient *preference elicitation* (as opposed to availability testing) protocols are developed to identify the winner.

Our model is also strongly related to *control via adding candidates* [2, 10], in which the initial candidates can be augmented by set of "spoiler" candidates, and a chair, with perfect knowledge of the votes, attempts to find a subset of spoiler candidates whose addition makes her preferred candidate win. We develop the connections to our model further in Sec. 3. Rastegari *et al.* [17] also look for optimal knowledge-gathering policies in a social choice context, although their setting and assumptions differ from ours. Their goal is to find a stable matching (e.g., between companies and job applicants), and knowledge-gathering actions (e.g., interviews of applicants) are intended to reduce uncertainty in the agents' rankings (e.g., a company's assessment of an applicant).

3 Computing Robust Winners

We first consider the problem of identifying or verifying robust winners given some available set. If x is a robust winner w.r.t. $\langle X, Y, \mathbf{v}, r\rangle$, it must meet two obvious necessary conditions: $x \in Y$ and $x = r(\mathbf{v})$ (obtained by setting $A = Y$ and $A = X$). We have the following key result:

Proposition 1 *Let r be a voting rule, $\mathbf{v}$ a profile over X, and $Y \subseteq X$ a set of available candidates. Candidate x is a robust winner w.r.t. $\langle X, Y, \mathbf{v}, r\rangle$ iff there is no destructive control against x by adding candidates, where the spoiler set is $U = X \setminus Y$.*

The proof is immediate: the chair has a destructive control against x via adding candidates iff there is a spoiler set $S \subseteq U$ such that $r(\mathbf{v}^{\downarrow Y \cup S}) \neq x$ (i.e., x is not a robust winner). Since the robust winner problem is equivalent to the complement of the problem of DESTRUCTIVE CONTROL BY ADDING CANDIDATES, results in the literature on election control directly determine the complexity of checking the existence of robust winners for several voting rules: it is coNP-complete for plurality [2, 10], Bucklin [5] and ranked pairs [15]; and it is polynomial for Copeland [6] and maximin [7]. The latter two results come with efficient algorithms for the robust winner problem. These results suggest that the problem tends to be simpler for voting rules that are based on the (unweighted or weighted) majority graph, because majority preference between two candidates x, y does not depend on the availability of other candidates. We provide a simple characterization of robust winners for the top cycle rule (recall that we assume n odd):

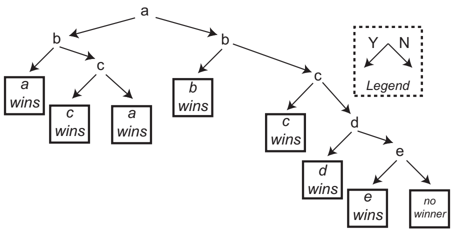
Proposition 2 *Let r be the top cycle rule. Candidate x is a robust winner w.r.t. $\langle X, Y, \mathbf{v}, r\rangle$ iff, for all $y \in X$, there is a path from x to y in $m(\mathbf{v})$ that goes only through candidates in Y.*[3]

As a corollary, checking whether x is a robust winner can be done in polynomial time.

Another interesting notion is that of an *irrelevant candidate*, which can be exploited in computing both robustness and optimal query policies (Sec. 4).

Definition 2 *Let $\mathbf{v}$ be a profile over X, $x \in X$, $Y \subseteq X \setminus \{x\}$ be the known available candidates, and r a voting rule. Candidate x is* irrelevant *w.r.t. $\langle \mathbf{v}, Y, r\rangle$ if for any $A \subseteq X \setminus (Y \cup \{x\})$, we have $r(\mathbf{v}^{\downarrow Y \cup A \cup \{x\}}) = r(\mathbf{v}^{\downarrow Y \cup A})$.*

[3]Proofs of results omitted for space reasons can be found in a longer version of the paper.



Vote profile:

$abcde$	$abcde$	$bcaed$
$bcead$	$adbec$	$cdeba$
$adcbe$	$cbade$	$cdbea$

Figure 1: A plurality-sufficient query policy for vote profile shown.

Notice that if x is irrelevant for Y, we need not consider the availability of x when testing the robustness of any candidate in $X \setminus (Y \cup \{x\})$ w.r.t. Y (or any superset of Y). Similarly, in any policy for determining winners, an availability test for an irrelevant x is useless once Y (or any superset) is known to be available, a fact we can exploit below. We have the following simple characterization of irrelevant candidates for a rich class of voting rules—informally, it says that once we know that at least one candidate in the top cycle is available, removing any candidate that is not in the top cycle cannot impact the choice of winner.

Proposition 3 *Let r be a voting rule satisfying the following property: for any profile $\mathbf{v}$, if $x \notin TC(\mathbf{v})$ then $r(\mathbf{v}^{\downarrow X \setminus \{x\}}) = r(\mathbf{v})$. For any $\mathbf{v}$, any $Y \subseteq X$ s.t. $Y \cap TC(\mathbf{v}) \neq \emptyset$, and any $x \in X \setminus TC(\mathbf{v})$, x is irrelevant w.r.t. $\langle \mathbf{v}, Y, r \rangle$.*

Prop. 3 applies, in particular, to the top cycle, Copeland, Slater and Banks rules.

4 Minimizing Expected Query Cost

We now consider the problem of computing optimal query policies that determine enough information about candidate availability to be able to declare a (robust) winner. We formulate the problem as one of *cost-sensitive decision tree construction*, present some experiments comparing the performance of myopic heuristics to optimal dynamic programming, and discuss a number of interesting theoretical and practical directions for future research.

4.1 Optimal Query Policies

A *query policy* is a binary decision tree T in which each non-leaf node n is labeled by a query $q(n) \in U$, the two outgoing edges from non-leaf node n are labeled by responses yes (or "available") and no (or "unavailable"), and each leaf l is labeled by an element $w(l)$ of X (the proposed *winner* given the query-response path to l). Let $yes(n)$ and $no(n)$ denote the yes/no successors of node n in T. Any path from the root of T to a leaf l induces the obvious information set $Q(l) = \langle Q^+(l), Q^-(l) \rangle$. A policy/tree T is *r-sufficient* w.r.t. $\mathbf{v}$ if: (a) the information set $Q(l)$ for each leaf l is r-sufficient; and (b) each leaf l is labeled with the robust winner $w(Q(l))$. For any $A \in \mathcal{A}$, let $l(A)$ denote the (unique) leaf that will be reached when responses are dictated by A, and $\pi(A)$ the corresponding path. Fig. 1 shows a vote profile and an r-sufficient tree for plurality voting.

The *query cost* of a (complete) path π in T is $c(\pi) = \sum_{x \in \pi} c(x)$, i.e., the sum of the costs of the queries on π. The *expected query cost* $c(T)$ of policy T is simply $E_{A \sim P} c(\pi(A))$. This can be computed in bottom up fashion as follows, with $c(T)$ being the cost of the root of T.

$$c(l) = 0 \text{ for leaf } l;$$
$$c(n) = c_{q(n)} + p_{q(n)} c(yes(n)) + (1 - p_{q(n)}) c(no(n)) \text{ for non-leaf } n.$$

If all candidates are available with probability $p = 0.9$, the tree in Fig. 1 has an expected query cost of 2.10. Our aim is to find a minimal cost, r-sufficient (i.e., optimal) policy T:

$$\arg\min\{c(T) : T \text{ is } r\text{-sufficient for } \mathbf{v}\}.$$

The problem of computing a minimal cost policy can be cast as a standard decision tree construction (or induction) problem [16]. We can view every available set $A \in \mathcal{A}$ as a training example labeled with its winner $w(A)$. We encode A using a binary feature vector $f(A)$, where each feature corresponds to the presence or absence of a specific candidate in U, and label it with associated probability $P(A)$. Thus our initial set of training examples is simply

$$\{\langle A, w(A), P(A)\rangle : A \in \mathcal{A}\},$$

where each A is encoded by a binary feature vector of length $|U|$.

Clearly the set of training examples has exponential size, requiring winner computation for exponentially many "elections" (of various sizes). Even if winner determination for a fixed candidate set is easy (i.e., polynomial time) for the voting rule in question, simply constructing the inputs for DP (or our myopic decision tree approach to follow) will be difficult.[4] We discuss ways to prune the set of training examples below. Cost-optimal decision trees can be computed readily using dynamic programming (DP) [8] or branch-and-bound [13]. For general binary classification, the (decision variant of) the problem of computing optimal decision trees is NP-complete, even with uniform probabilities and costs [11].[5] However, given the importance of minimizing query costs, we believe even intense computation will be worthwhile.

Standard DP for decision trees structures the problem by considering *sets of training examples* $E \subseteq \mathcal{E} = 2^{\mathcal{A}}$. A set E is *pure* if all examples in E are labeled with the same winner. The *optimal cost function* c^* for an arbitrary example set $E \subseteq \mathcal{E}$ is the cost of the optimal policy knowing initially that the available set A lies within E:

$$c^*(E) = \begin{cases} 0 & \text{if } E \text{ is pure} \\ \min_{q \in U} p(q^+|E)c^*(E_q^+) + p(q^-|E)c^*(E_q^-) + c_q & \text{if } E \text{ is not pure} \end{cases}$$

Since $c^*(E)$ depends only on the optimal optimal costs for subsets of E, DP can be used, computing optimal costs for smaller sets before larger ones.

Naïve DP is doubly exponential in the size of the candidate set U: $|\mathcal{E}| = 2^{|\mathcal{A}|} = 2^{2^{|U|}}$. But the structure in our problem gives us very restricted subsets of training examples. First, observe that example set E_0 at the root consists of all subsets A (i.e., $E_0 = \mathcal{A}$). Second, every query-response path of length k gives a set E equal to E_0 restricted to the instantiation of k queries, i.e., the set of feature vectors where k positions are fixed to some k-bit vector, and all $2^{|U|-k}$ instantiations of the remaining $|U| - k$ features are present. Thus the only realizable sets E have size $2^{|U|-k}$, for some $0 \leq k \leq |U|$, and each such E has the form described above. Hence, the number of "reachable" example sets required for DP is exponential rather than doubly exponential:

$$\sum_{k=0}^{|U|} 2^k \binom{|U|}{k} = 3^{|U|}.$$

Thus the optimal query policy can be computed in $O(3^{|U|})$ time for any profile and any voting rule.

The DP approach is generic and exploits no structure at all in the profile, nor any special properties of the voting rule itself. An important research direction is the refinement of the DP algorithm for specific voting rules in a way that could (perhaps drastically) prune the subsets A that need to be explicitly considered. For example, for rules such as top cycle, Copeland, Slater and Banks, Prop. 3

[4] Indeed, it will be NP-hard in general; we thank one of the reviewers for this observation.

[5] We are exploring reductions to confirm the complexity of our problem, which we believe to be PSPACE-complete.

allows us to "collapse" potentially large numbers of candidate subsets—those that vary only in the availability of "irrelevant candidates"—and treat them as a single training example with a unique winner. Similarly, rules that satisfy the "majority winner property"—i.e., if a candidate is in the first position of the majority of votes, it must win—allows significant pruning as well: in any subset Y with such a majority winner x, x's availability renders all remaining candidates irrelevant. This type of pruning has two direct benefits: it reduces both the (explicit) set $\mathcal{A}$ (i.e., number of training examples), and the number of subsets E of $\mathcal{A}$ that must be considered (as we discuss below).

4.2 Myopic Query Tree Construction

Because of the NP-hardness of optimal decision tree construction in general, and the exponential complexity of DP in particular, the machine learning community has considered heuristic, *myopic* approaches to decision tree induction, one of the most popular being C4.5, which is based on *information gain* [16]. Extensions to cost-sensitive classification have been considered as well [9, 18], and such schemes can be adapted to our setting easily. We now outline such an approach.

For any set of training examples $E \subseteq \mathcal{E}$, define $w(E)$ to be the set of winners $w(A)$ that occur in some example $A \in E$. Let $p_E(x) = \sum\{P(A) : A \in E, w(A) = x\}$ be the probability that x wins in training set E. The *entropy* of E is

$$H(E) = \sum_{x \in w(E)} -p_E(x) \log p_E(x).$$

A *split* of E on a feature (i.e., candidate, query) q partitions E into those examples $E_q^+ \subseteq E$ where q is available, and $E_q^- \subseteq E$ where q is not. The *conditional entropy* of training set E given q is:

$$H(E|q) = p(q^+|E)H(E_q^+) + p(q^-|E)H(E_q^+).$$

We define the *information gain* associated with query q to be:

$$IG(E|q) = H(E) - H(E|q).$$

Myopic decision tree construction proceeds as follows. We first initialize the tree with a single leaf (root) node n_0 and associate with n_0 the set $E_0 = \mathcal{A}$ all labeled training examples. Then we repeat the following operations on unprocessed nodes until no nodes are unprocessed. Let n be an unprocessed node and $E(n)$ be its associated training set:

1. If n is pure, designate it processed; label it with its (unique) winner (it now becomes a leaf).
2. If n is not pure, then:
   (a) For each (non-redundant) feature split $q \in U$, compute its information gain $IG(E(n)|q)$.
   (b) Split n using the query q with maximum information gain, creating children: $n_q^+ = yes(n)$ on the yes edge, associated with examples E_q^+; and $n_q^- = no(n)$ on the no edge, associated with examples E_q^-.
   (c) Designate n processed, and n_q^+ and n_q^- unprocessed.

Processing of any node is linear in the number of training examples at a node, hence the complexity of myopic decision tree induction is linear in (a possibly pruned) $\mathcal{A}$ and the size of the resulting tree. Since $\mathcal{A}$ has size $2^{|U|}$ in the worst case (if unpruned), complexity of myopic induction is $O(2^{|U|})$, i.e., significantly more efficient than optimal tree construction using DP (which has unpruned complexity $O(3^{|U|})$). This improved efficiency comes at a price, since the use of information gain is not guaranteed to result in a policy with minimal expected query cost. However, we expect it to work well in practice; and it *is guaranteed* to provide a *correct* policy (i.e., one that determines the true winner).

If we are willing to live with potential error in the declaration of the winner, other forms of approximation can be considered in constructing the policy. We describe two here, but leave their detailed investigation to future research. The approaches above ensure we always obtain the correct winner, since the policy tree has only pure leaves (i.e., all candidate sets A at a leaf have the same winner). One form of approximation is to terminate the querying process at *impure leaves*, requiring only that we be "sure enough" about the identity of the winner, and predicting a winner despite this residual uncertainty. This is analogous to cost-sensitive classification [9, 18], where both tests and and *prediction errors* have costs. In our setting, we have two distinct types of misclassification errors/costs: (a) if we predict/choose a winner who turns out to be unavailable; (b) if we predict/choose a winner that is available, but is not the true winner given the actual (unknown) available set. In general, we expect the former to be much worse than the latter.[6] An important research direction will be to to modify the query policy algorithms so these costs are taken into account.

Another important approximation, which can help reduce the number of training examples and render myopic decision tree construction fully tractable, is to build the tree using *sampled availability sets*, where training examples A are drawn from the distribution P over $\mathcal{A}$. With a constant number of sampled sets, decision tree induction becomes linear in the number of candidates and size of T. Sample complexity results then become an important research direction [9].

4.3 Extreme Availability Probabilities

When candidate availability probabilities are extreme, i.e., close to close to 1 or 0, constructing optimal query policies becomes much easier. Assume $p_x = p$ for all x and all query costs are identical. We first consider the case where p is very close to 1, reflecting settings where an unavailable candidate would be exceptional. Let $p = 1 - O(\varepsilon)$. Consider the following query policy $Extreme(\mathbf{v})$, which (informally) proceeds as follows. We initialize the *potential set* X to contain all candidates, the *known set* $Y = \emptyset$, and the *current winner* $w = r(\mathbf{v})$. We then repeat the following steps:

1. look for a minimum-cardinality subset Z of $X \setminus Y$ such that w is a robust winner for $Y \cup Z$ (note that Z must contain w if w is not known to be available);
2. check the availability of all candidates in Z;
3. if all candidates in Z are available, stop and output w;
4. if not, remove the unavailable candidates from the profile $\mathbf{v}$, recompute $w = r(\mathbf{v})$, and go back to step 1.

As an example, let r be plurality and consider the profile $\mathbf{v}$ shown in Fig. 1. The (initial) current winner $r(\mathbf{v})$ is a. We then have $Z = \{a, b\}$. Under our extreme availability assumption, with high probability we will learn that both a and b are available, in which case we stop and output a. However, suppose that we learn a is available but b is not. We would then replace $\mathbf{v}$ by $\mathbf{v}^{\downarrow\{a,c,d,e\}}$, giving a new current winner c, and $Z = \{c\}$. After checking Z, suppose we learn c is not available; we then replace $\mathbf{v}$ by $\mathbf{v}^{\downarrow\{a,d,e\}}$, giving new current winner a, and $Z = \emptyset$: we stop and output a.

It is not hard to show that $Extreme(\mathbf{v})$ terminates, and returns the true winner $r(\mathbf{v}^{\downarrow A})$ for the actual available set A, provided at least one candidate is actually available. Now suppose $Y \subseteq X$ is a smallest (cardinality) set of candidates such that $r(\mathbf{v})$ is a robust winner for Y, and $|Y| = k$. Then we have that the expected cost of the policy is $k + O(\varepsilon)$. We can also show that any query policy must have this expected cost:

[6] For instance, going to a restaurant without confirming availability, then arriving to find out it has no space, is worse than going to that same restaurant having obtained a reservation and then finding out another restaurant might have been preferable because some others became unavailable.

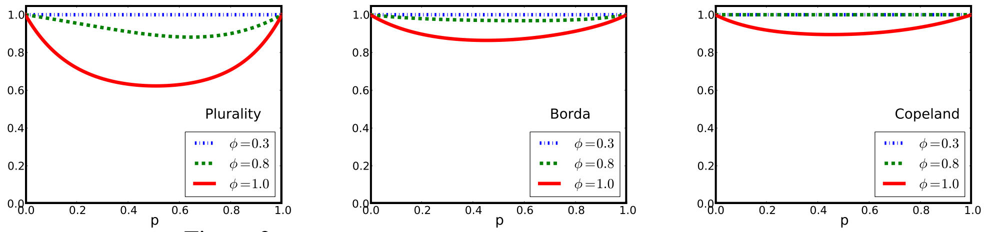


Figure 2: Probability an available naïve winner is the true winner.

Lemma 1 *Any r-sufficient query policy has an expected query cost of at least $k - O(\varepsilon)$.*

Proof: With probability $p^{|X|}$, all candidates are available. When all candidates are available, $x = r(\mathbf{v})$ is the true winner, and to verify it we must check the availability of $k-1$ other candidates (this being the smallest size set that admits a robust winner). Therefore, the expected cost of any query policy is at least $p^{|X|} \cdot k = k - O(\varepsilon)$. ∎

These facts together directly demonstrate the following:

Proposition 4 *The policy $Extreme(\cdot)$ is asymptotically optimal as $\varepsilon \to 0$.*

The opposite case, where all candidates are highly unlikely to be available, is much less interesting: it is very likely that we must check the availability of all candidates—even then it is highly likely that no candidate is available—and run the voting on the available set. The naïve policy that tests all candidates in advance of voting is (asymptotically) optimal.

4.4 Empirical Evaluation

We now describe some simple experiments designed to test the effectiveness of our algorithms for computing query policies, and examine the expected costs of these policies for various voting rules, preference distributions and availability distributions. In all of the experiments that follow, we generate vote profiles using *Mallows distributions* over rankings [14]. The Mallows model is a distribution over rankings given by a modal ranking σ over X and dispersion $\phi \in (0, 1]$: the probability of vote v is $\Pr(v|\sigma, \phi) \propto \phi^{d(r,\sigma)}$, where d is Kendall's τ-distance (each vote is drawn independently). Smaller ϕ concentrates mass around σ while $\phi = 1$ gives the uniform distribution, i.e., Mallows with $\phi = 1$ corresponds precisely to the *impartial culture assumption*. In all experiments, we consider $m = 10$ candidates and $n = 100$ voters. We generate vote profiles for $\phi \in \{0.3, 0.8, 1.0\}$, and consider three different voting rules: plurality, Borda and Copeland. We consider various availability probabilities p, which vary depending on the circumstance, but in all cases each candidate has the same availability probability. Results for each problem instance (combination of voting rule, ϕ, p combination) are reported as averages (and other statistics) over 25 randomly drawn vote profiles.

Before exploring the performance of query policies, we first measure the probability of error (i.e., selecting an incorrect winner) associated with a naïve policy which simply selects the *naïve winner*, $r(\mathbf{v})$, without regard to candidate unavailability. Obviously, a lower bound on this error is $1-p$, since the winner will be unavailable with that probability. Fig. 2 shows this error probability *conditional on the winner being available* for the three voting rules considered, for different ϕ, as we vary the availability probability p. For p near 1, the naïve winner is, of course, almost always correct. At the other extreme, when candidates are usually not available, the naïve winner is usually correct also, since it is highly likely to be the only available option. We see that when preferences are very peaked ($\phi = 0.3$), candidate deletion has little impact, since most voters rank all candidates very similarly; but as preferences become more uniform—in particular, for impartial culture ($\phi = 1$)—the impact is dramatic, especially for plurality, and somewhat less so for Copeland. This suggests that testing availability is important even for reasonably high availability probabilities. It is important to

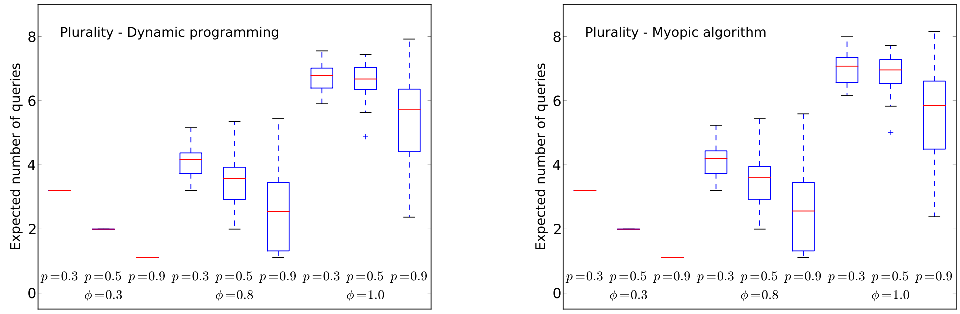


Figure 3: Expected query cost of decision trees for plurality using (a) dynamic programming and (b) the information gain heuristic.

ϕ	0.3			0.8			1.0		
p	0.3	0.5	0.9	0.3	0.5	0.9	0.3	0.5	0.9
Plur-DP	3.2 (3.2,3.2)	2.0 (2.0,2.0)	1.1 (1.1,1.1)	4.1 (3.2,5.2)	3.4 (2.0,5.4)	2.7 (1.1,5.4)	6.7 (5.9,7.6)	6.6 (4.9,7.4)	5.4 (2.4,7.9)
Borda-DP	3.2 (3.2,3.2)	2.0 (2.0,2.0)	1.1 (1.1,1.1)	3.7 (3.2,4.5)	2.7 (2.0,3.9)	1.7 (1.1,5.0)	5.4 (4.4,6.7)	4.8 (3.2,6.4)	3.3 (1.2,6.2)
Cope-DP	3.2 (3.2,3.2)	2.0 (2.0,2.0)	1.1 (1.1,1.1)	3.2 (3.2,3.6)	2.0 (2.0,2.5)	1.1 (1.1,1.3)	4.6 (3.4,5.9)	3.6 (2.1,5.6)	2.2 (1.1,4.5)
Plur-IG	3.2 (3.2,3.2)	2.0 (2.0,2.0)	1.1 (1.1,1.1)	4.1 (3.2,5.2)	3.5 (2.0,5.5)	2.8 (1.1,5.6)	7.0 (6.2,8.0)	6.9 (5.0,7.7)	5.6 (2.4,8.2)
Borda-IG	3.2 (3.2,3.2)	2.0 (2.0,2.0)	1.1 (1.1,1.1)	3.7 (3.2,4.6)	2.7 (2.0,3.9)	1.7 (1.1,5.0)	5.5 (4.5,7.0)	4.9 (3.2,6.7)	3.3 (1.2,6.2)
Cope-IG	3.2 (3.2,3.2)	2.0 (2.0,2.0)	1.1 (1.1,1.1)	3.2 (3.2,3.6)	2.0 (2.0,2.5)	1.1 (1.1,1.3)	4.7 (3.5,6.7)	3.7 (2.1,5.9)	2.2 (1.1,4.5)

Table 1: Avg. query cost (min, max) for optimal (DP) and myopic (IG) query policies.

realize that these plots give only a sense of the "degree of robustness" of a winner *who is assumed to be available*, even for low p (where the odds of winner availability is low): as such, they do not necessarily provide insight into the value of intelligent availability testing.

We now consider the expected number of queries required to determine the winner under the scenarios described above (using the different values of ϕ) under three different availability probabilities: $p = 0.3, 0.5, 0.9$. Fig. 3 plots expected query cost for plurality, using both dynamic programming (which gives the optimal policy) and the myopic information gain heuristic, for all nine parameter settings. The plots show average expected cost, standard error, and maximum and minimum expected costs over 25 trials. The results for all three rules are shown in Table 1.

The first thing to notice is that, in most settings, the use of optimal query policies can offer significant savings in availability tests relative to the standard approach of first testing the availability of all ten candidates. We see that the myopic heuristic tends to produce trees with costs very close to the optimum: even in the problems instances with the largest average gap (i.e., for plurality with $\phi = 1$), the myopically constructed trees have an expected cost of only 0.3 more queries than optimal on average; in most cases, these trees are almost identical to the corresponding optimum. This suggests that the more efficient myopic approaches will work well at minimizing availability query costs in practice. Not surprisingly, we see strong (negative) correlations between query costs and availability probability in all three voting rules. The query cost is also correlated with dispersion ϕ: when ϕ is greater (more uniform) query costs are higher since preferences are more diverse. When dispersion ϕ is low, given a fixed p, expected query cost is the essentially the same for each voting rule, and the myopic algorithm produces virtually optimal policies (indeed, identical in cost to the optimum up to the reported precision).

Table 2 shows the sizes of the decision trees that result when running both of our algorithms: tree size is only indirectly related to expected query cost, since the relative balance of the trees also impacts expected query costs. Nonetheless we see an expected correlation, though we notice that plurality tends to result in larger trees, especially for $\phi = 1$. We also see that the myopic trees are not significantly larger than the optimal trees, though the differences in size are somewhat more pronounced than the differences in expected query cost described above.

ϕ	0.3			0.8			1.0		
p	0.3	0.5	0.9	0.3	0.5	0.9	0.3	0.5	0.9
Plur-DP	9.0 (9,9)	9.0 (9,9)	9.0 (9,9)	52.0 (9,128)	49.6 (9,124)	57.0 (9,148)	233.8 (133,311)	221.2 (121,302)	249.6 (136,318)
Borda-DP	9.0 (9,9)	9.0 (9,9)	9.0 (9,9)	24.4 (9,61)	24.0 (9,57)	26.2 (9,63)	114.8 (42,209)	110.3 (41,197)	125.1 (44,226)
Cope-DP	9.0 (9,9)	9.0 (9,9)	9.0 (9,9)	10.3 (9,19)	10.3 (9,19)	10.3 (9,19)	58.4 (17,161)	57.8 (17,160)	63.1 (17,185)
Plur-IG	9.0 (9,9)	9.0 (9,9)	9.0 (9,9)	59.5 (9,140)	55.4 (9,140)	62.2 (9,163)	290.8 (163,379)	258.6 (145,351)	296.9 (171,402)
Borda-IG	9.0 (9,9)	9.0 (9,9)	9.0 (9,9)	26.8 (9,63)	24.1 (9,59)	26.5 (9,68)	136.5 (49,264)	117.8 (42,229)	135.1 (46,253)
Cope-IG	9.0 (9,9)	9.0 (9,9)	9.0 (9,9)	10.3 (9,19)	10.3 (9,19)	10.4 (9,20)	67.1 (21,211)	60.8 (18,178)	70.2 (18,213)

Table 2: Avg. tree size (min, max) for optimal (DP) and myopic (IG) (number of query nodes).

4.5 Query Complexity

Apart from optimizing query policies, the theoretical question of both worst-case query (availability test) complexity, and average-case query complexity for specific distributions, is of interest. Here we sketch some partial results just to offer a flavor for the types of questions one might ask in the unavailable candidate model under our availability testing regime.

Worst-case results take the form: given a voting rule r and availability distribution P, what is the greatest (over possible vote profiles $\mathbf{v}$) expected (over availabilities) query cost of the optimal query policy? Distributions where candidates are highly likely to be available allow one to construct profiles where determining the winner requires almost m queries in expectation, for both plurality and Borda. Our constructions partition votes into two sets each with a distinct "winning" candidate in each set with equal scores (and arranging other candidates appropriately "uniformly" so that a precise subset of available candidates is needed to determine which of the two top candidates wins). The hardness of these cases lies in the high concentration of the binomial distribution. In the extreme case where p is very close to 1, for any voting rule the expected query complexity is

$$\min\{|Y| \text{ s.t. } w = r(\mathbf{v}) \text{ is a robust winner w.r.t. } (r, X, Y)\} + O(\varepsilon)$$

(this can be seen from our earlier construction). By contrast, if p is very close to 0, it is highly likely that we must check the availability of every candidate, giving an expected query complexity $m + O(\varepsilon)$.

With regard to expected query complexity, interesting questions arise when considering the expected optimal query cost not for worst-case profiles, but on average for preference or vote profiles drawn from particular distributions, such as impartial culture, various forms of Mallows models or mixtures, distributions over single-peaked preferences, etc., for various voting rules.

5 Future Directions

We have offered a new perspective on voting in the unavailable candidate model, assuming that testing the viability or availability of candidates is costly. We have presented several new concepts, including those of robust winners, irrelevant candidates, and availability/query policies, and provided algorithms for the computation of both optimal and myopic query policies by exploiting connections to decision tree induction. Our experimental results with plurality, Borda and Copeland voting show the value of optimal querying, with significant savings (relative to determining availability of all candidates) realized over a variety of preference and availability distributions.

There are of course a variety of interesting directions to be pursued in this nascent line of research. Many of these have been suggested above, but we summarize some of them here. A critical direction of both practical and theoretical interest is developing efficient methods for pruning the available sets used in policy construction based on specific properties of voting rules. Heuristic methods for constructing policies could be useful; e.g., we might exploit the probability a given candidate will win—as in the control problem addressed in [19]—to determine the next candidate to query. Sample-based procedures, where only some available sets are classified, may also prove to be important in minimizing computational costs. Other representations of winner determination policies (e.g., decision lists or graphs) may be of value. Extensions to policies that "predict" winners, rather than guaranteeing robustness—whether in a *speculative* (choosing winners that may not be available) or *safe* (only selecting known available candidates) fashion—is of great interest. We are exploring some of the query and communication complexity questions mentioned above. Finally, the question of new opportunities for manipulation under this model seem rather intriguing.

Acknowledgments: Thanks to the reviewers for helpful suggestions. This work was supported in part by NSERC, by project ComSoc (ANR-09-BLAN-0305-01), and by MICINN project TIN2011-27652-C03-02.

References

[1] K. Baldiga and J. Green. Assent-maximizing social choice. *Social Choice and Welfare*, online first, 2011.

[2] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election? *Social Choice and Welfare*, 16(8-9):27–40, 1992.

[3] Y. Chevaleyre, J. Lang, N. Maudet and J. Monnot. Possible winners when new candidates are added: the case of scoring rules. *Proceedings of AAAI-10*, 2010.

[4] B. Dutta, M. Jackson and M. Le Breton. Strategic candidacy and voting procedures. *Econometrica*, 69(4):1013–1037, 2001.

[5] G. Erdélyi, M. Fellows, L. Piras and J. Rothe. Control complexity in Bucklin and fallback voting. arXiv 1103.2230, 2011.

[6] P. Faliszewski, E. Hemaspaandra and H. Schnoor. Copeland voting: ties matter. *AAMAS-08*, pp. 983-990, 2008

[7] P. Faliszewski, E. Hemaspaandra and L. Hemaspaandra Multimode control attacks on elections. *JAIR*, 40:305–351, 2011.

[8] M. R. Garey. Optimal binary identification procedures. *SIAM J. Appl. Math.*, 23:173–186, 1972.

[9] R. Greiner, A. J. Grove and D. Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139(2):137–174, 1992.

[10] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5-6):255–285, 2007.

[11] L. Hyafil and R. Rivest Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5:15–17, 1976.

[12] T. Lu and C. Boutilier. The unavailable candidate model: a decision-theoretic view of social choice. *ACM Conference on Electronic Commerce 2010*, pp.263–274, 2010.

[13] A. Martelli and U. Montanari. Optimizing decision trees through heuristically guided search. *Communications of the ACM*, 21(12):1025–1039, 1978.

[14] C. L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

[15] D. Parkes and L. Xia. A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. AAAI-12, to appear, 2012.

[16] J. R. Quinlan. *C45: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[17] B. Rastegari, A. Condon, K. Leyton-Brown and N. Immorlica. Two-Sided matching with partial information. Working paper, 2011.

[18] P. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409, 1995.

[19] K. Wojtas and P. Faliszewski. Possible winners in noisy elections. AAAI-12, to appear, 2012.

Craig Boutilier
Department of Computer Science
University of Toronto
Email: `cebly@cs.toronto.edu`

Jérôme Lang
LAMSADE
Université Paris-Dauphine
Email: `lang@lamsade.dauphine.fr`

Joel Oren
Department of Computer Science
University of Toronto
Email: `oren@cs.toronto.edu`

Héctor Palacios
Departamento de Informática
Universidad Carlos III de Madrid
Email: `hpalacio@inf.uc3m.es`

A characterization of the single-crossing domain

Robert Bredereck, Jiehua Chen, and Gerhard J. Woeginger

Abstract

We characterize single-crossing preference profiles in terms of two forbidden substructures, one of which contains three voters and six (not necessarily distinct) alternatives, and one of which contains four voters and four (not necessarily distinct) alternatives. We also provide an efficient way to decide whether a preference profile is single-crossing.

1 Introduction

Restricted domains of preferences. Single-peaked and single-crossing preferences have become standard domain restrictions in many economic models. Preferences are *single-peaked* if there exists a linear ordering of the alternatives such that any voter's preference relation along this ordering is either always strictly increasing, always strictly decreasing, or first strictly increasing and then strictly decreasing. Preferences are *single-crossing* if there exists a linear ordering of the voters such that for any pair of alternatives along this ordering, there is a single spot where the voters switch from preferring one alternative above the other one. In many situations, these assumptions guarantee the existence of a strategy-proof collective choice rule, or the existence of a Condorcet winner, or the existence of an equilibrium.

Single-peaked preferences go back to the work of Black [5] and have been studied extensively over the years. Single-peakedness implies a number of nice properties, as for instance non-manipulability (Moulin [19]) and transitivity of the majority rule (Inada [14]). Single-crossing preferences go back to the seminal paper of Roberts [20] on income taxation. Grandmont [12], Rothstein [21], and Gans & Smart [11] analyze various aspects of the majority rule under single-crossing preferences. Furthermore, single-crossing preferences play a role in the areas of income redistribution (Meltzer & Richard [18]), coalition formation (Demange [8]; Kung [15]), local public goods and stratification (Westhoff [24]; Epple & Platt [9]), and in the choice of constitutional voting rules (Barberà & Jackson [3]). Saporiti & Tohmé [23] study single-crossing preferences in the context of strategic voting and the median choice rule, and Saporiti [22] investigates them in the context of strategy proof social choice functions. Barberà & Moreno [4] develop the concept of top monotonicity as a common generalization of single-peakedness and single-crossingness (and of several other domain restrictions).

Forbidden substructures. Sometimes mathematical structures allow characterizations through forbidden substructures. For example, Kuratowski's theorem [16] characterizes planar graphs in terms of forbidden subgraphs: a graph is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$. For another example, Hoffman, Kolen & Sakarovitch [13] characterize totally-balanced 0-1-matrices in terms of certain forbidden submatrices. In a similar spirit, Lekkerkerker & Boland [17] characterize interval-graphs through five (infinite) families of forbidden induced subgraphs.

In the area of social choice, a beautiful result by Ballester & Haeringer [2] characterizes single-peaked preference profiles in terms of two forbidden substructures. The first forbidden substructure concerns three voters and three alternatives, where each of the voter ranks another one of the alternatives worst. The second forbidden substructure concerns two voters and four alternatives, where (sloppily speaking) both voters rank the first three alternatives in opposite ways with the second alternative in the middle, but prefer the fourth alternative to the second one.

Contribution of this paper. Inspired by the approach and by the results of Ballester & Haeringer [2], we present a forbidden substructure characterization of single-crossing preference profiles. One of our forbidden substructures consists of three voters and six alternatives (as described in Example 2.4) and the other one consists of four voters and four alternatives (as described in Example 2.5). We stress that the (six respectively four) alternatives in these forbidden substructures are not necessarily distinct: the substructures only partially specify the preferences of the involved voters; hence by identifying and collapsing some of the involved alternatives we can easily generate a number of smaller forbidden substructures (which of course are just special cases of our larger forbidden substructures). Finally, we will discuss the close relation of single-crossing preference profiles to *consecutive ones matrices*. A 0-1-matrix has the consecutive ones property if its columns can be permuted such that the 1-values in each row are consecutive. We hope that our results will turn out useful for future research on single-crossing profiles.

In Section 2 we summarize the basic definitions and provide some examples. In Section 3 we formulate and prove our main result (Theorem 3.1). In Section 4 we discuss the tightness of our characterization, and we argue that there does not exist a characterization that works with smaller forbidden substructures. Finally in Section 5 we show how to recognize the single-crossing property in polynomial time by using the connection to consecutive ones matrices.

2 Definitions, notations, and examples

Let $a_1, \ldots, a_m$ be m alternatives and let $V_1, \ldots, V_n$ be n voters. A *preference profile* specifies the *preference orderings* of the voters, where voter V_i ranks the alternatives according to a strict linear order $\succ_i$. For alternatives a and b, the relation $a \succ_i b$ means that voter V_i strictly prefers a to b.

An unordered pair of two distinct alternatives is called a *couple*. A subset $\mathcal{V}$ of the voters is *mixed* with respect to couple $\{a, b\}$, if $\mathcal{V}$ contains two voters one of which ranks a above b, whereas the other one ranks a below b. If $\mathcal{V}$ is not mixed with respect to $\{a, b\}$, then it is said to be *pure* with respect to $\{a, b\}$. Hence, an empty set of voters is pure with respect to any pair of alternatives. A couple $\{a, b\}$ *separates* two sets $\mathcal{V}_1$ and $\mathcal{V}_2$ of voters from each other, if no voter in $\mathcal{V}_1$ agrees with any voter in $\mathcal{V}_2$ on the relative ranking of a and b; in other words, sets $\mathcal{V}_1$ and $\mathcal{V}_2$ must both be pure with respect to $\{a, b\}$, and if both are non-empty then their union $\mathcal{V}_1 \cup \mathcal{V}_2$ is mixed.

An ordering of the voters is *single-crossing with respect to couple* $\{a, b\}$, if the ordered list of voters can be split into an initial piece and a final piece that are separated by $\{a, b\}$. An ordering of the voters is *single-crossing*, if it is single-crossing with respect to every possible couple. Finally a preference profile is single-crossing, if it allows a single-crossing ordering of the voters. It is easy to see that single-crossing is a monotone property of preference profiles:

Lemma 2.1 *Let $\mathcal{P}$ be a preference profile, and let $\mathcal{P}'$ result from $\mathcal{P}$ by removing some alternatives and/or voters. If $\mathcal{P}$ is single-crossing, then also $\mathcal{P}'$ is single-crossing.* □

In the remaining part of this section we present several instructive examples of preference

V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}
1	1	1	1	5	5	5	5	5	5	5
2	2	2	5	1	1	1	4	4	4	4
3	3	5	2	2	2	4	1	1	3	3
4	5	3	3	3	4	2	2	3	1	2
5	4	4	4	4	3	3	3	2	2	1

Figure 1: A single-crossing preference profile with 11 voters and 5 alternatives.

profiles that are single-crossing (Section 2.1) respectively that are not single-crossing (Section 2.2).

2.1 Profiles from weak Bruhat orders

Let S_m denote the set of permutations of $1, \ldots, m$. We specify permutations $\pi \in S_m$ by listing the entries as $\pi = \langle \pi(1), \pi(2), \ldots, \pi(n) \rangle$. The *identity* permutation $\langle 1, 2, \ldots, m \rangle$ arranges the integers in increasing order, and the *order reversing* permutation $\langle m, m-1, \ldots, 2, 1 \rangle$ arranges them in decreasing order. A *descent* in π is a pair $(\pi(i), \pi(i+1))$ of consecutive entries with $\pi(i) > \pi(i+1)$. We write $\pi \lhd \rho$, if permutation π can be obtained from permutation ρ by a series of swaps, each of which interchanges the two elements of a descent.

The partially ordered set $(S_m, \lhd)$ is known as *weak Bruhat order*; see for instance Bóna[6]. The weak Bruhat order has the identity permutation as minimum element and the order reversing permutation as maximum element. Every maximal chain (that is: every maximal subset of pairwise comparable permutations) in the weak Bruhat order has length $\frac{1}{2}m(m-1)+1$ and contains the identity permutation and the order reversing permutation.

The following example illustrates the well-known connection between weak Bruhat orders and single-crossing preference profiles; we refer the reader to Abello [1] or Galambos & Reiner [10] for more information.

Example 2.2 *Let $C = (\pi_1 \lhd \pi_2 \lhd \cdots \lhd \pi_n)$ be a maximal chain with $n = \frac{1}{2}m(m-1)+1$ permutations in the weak Bruhat order $(S_m, \lhd)$. We construct a profile by using $1, \ldots, m$ as alternatives, and by interpreting every permutation π as preference ordering $\pi(1) \succ \pi(2) \succ \ldots \succ \pi(n)$ over the alternatives. Voter V_i has preference ordering π_i. See Figure 1 for an illustration with $m = 5$ alternatives and $n = 11$ voters.*

The resulting profile is single-crossing: any two alternatives a and b start off in the right order in the identity permutation π_1, eventually are swapped into the wrong order, and then can never be swapped back again at later steps. Furthermore, the profile contains $n = \frac{1}{2}m(m-1)+1$ voters with pairwise distinct preference orderings. □

If one starts the construction in Example 2.2 from arbitrary (not necessarily maximal!) chains in the weak Bruhat order, then one can generate this way every possible single-crossing preference profile (up to isomorphism). This is another well-known connection, which follows from the fact that $\pi \lhd \rho$ if and only if every inversion of permutation π also is an inversion of permutation ρ.

2.2 Some profiles that are not single-crossing

We next present three examples of profiles that are not single-crossing. The first example is due to Saporiti & Tohmé [23] and shows a profile that is single-peaked but fails to be single-crossing. The other two examples introduce two principal actors of this paper.

Example 2.3 *Consider four alternatives* $1, 2, 3, 4$ *and three voters* V_1*,* V_2*,* V_3 *with the following preference orders:*

$$\begin{aligned}
&\text{Voter } V_1\text{:} \quad 2 \succ_1 3 \succ_1 4 \succ_1 1\\
&\text{Voter } V_2\text{:} \quad 4 \succ_2 3 \succ_2 2 \succ_2 1\\
&\text{Voter } V_3\text{:} \quad 3 \succ_3 2 \succ_3 1 \succ_3 4
\end{aligned}$$

It can be verified that this profile is not single-crossing but single-peaked (with respect to the ordering $1 < 2 < 3 < 4$ *of alternatives, for instance).* □

Example 2.4 (γ-Configuration)
A profile with three voters V_1*,* V_2*,* V_3 *and six (not necessarily distinct) alternatives* a, b, c, d, e, f *is a* γ*-configuration, if it satisfies the following:*

$$\begin{aligned}
&\text{Voter } V_1\text{:} \quad b \succ_1 a \quad \text{and} \quad c \succ_1 d \quad \text{and} \quad e \succ_1 f\\
&\text{Voter } V_2\text{:} \quad a \succ_2 b \quad \text{and} \quad d \succ_2 c \quad \text{and} \quad e \succ_2 f\\
&\text{Voter } V_3\text{:} \quad a \succ_3 b \quad \text{and} \quad c \succ_3 d \quad \text{and} \quad f \succ_3 e
\end{aligned}$$

This profile is not single-crossing, as none of the three voters can be arranged between the other two: the couple $\{a, b\}$ *prevents us from putting* V_1 *into the middle, the couple* $\{c, d\}$ *forbids voter* V_2 *in the middle, and the couple* $\{e, f\}$ *forbids* V_3 *in the middle.* □

The observations stated in Example 2.4 provide a cheap proof that the profile in Example 2.3 is not single-crossing, as this profile contains a γ-configuration with $a = 3$, $b = c = 2$, $d = e = 4$, and $f = 1$.

Example 2.5 (δ-Configuration)
A profile with four voters V_1*,* V_2*,* V_3*,* V_4 *and four (not necessarily distinct) alternatives* a, b, c, d *is a* δ*-configuration, if it satisfies the following:*

$$\begin{aligned}
&\text{Voter } V_1\text{:} \quad a \succ_1 b \quad \text{and} \quad c \succ_1 d\\
&\text{Voter } V_2\text{:} \quad a \succ_2 b \quad \text{and} \quad d \succ_2 c\\
&\text{Voter } V_3\text{:} \quad b \succ_3 a \quad \text{and} \quad c \succ_3 d\\
&\text{Voter } V_4\text{:} \quad b \succ_4 a \quad \text{and} \quad d \succ_4 c
\end{aligned}$$

This profile is not single-crossing: the couple $\{a, b\}$ *forces us to place* V_1 *and* V_2 *next to each other, and to put* V_3 *and* V_4 *next to each other; the couple* $\{c, d\}$ *forces us to place* V_1 *and* V_3 *next to each other, and to put* V_2 *and* V_4 *next to each other. Then no voter can be put into the first position.* □

3 A characterization through forbidden configurations

Examples 2.4 and 2.5 demonstrate that preference profiles that contain a γ-configuration or a δ-configuration cannot be single-crossing. It turns out that these two configurations are the only obstructions for the single-crossing property.

Theorem 3.1 *A preference profile* $\mathcal{P}$ *is single-crossing if and only if* $\mathcal{P}$ *contains neither a* γ*-configuration nor a* δ*-configuration.*

The rest of this section is dedicated to the proof of Theorem 3.1. The (only if) part immediately follows from the monotonicity of the single-crossing property (Lemma 2.1) and from the observations stated in Examples 2.4 and 2.5.

For the (if) part, we first introduce some additional definitions and notations. An *ordered partition* $\langle X_1, \ldots, X_p \rangle$ of the voters $V_1, \ldots, V_n$ satisfies the following properties: every part X_i is non-empty, distinct parts are disjoint, and the union of all parts is the set of all voters. The *trivial* ordered partition has $p = 1$ and hence consists of a single part $\{V_1, \ldots, V_n\}$. We let $\{a_k, b_k\}$ with $1 \leq k \leq \frac{1}{2}m(m-1)$ be an enumeration of all the possible couples, and we define $\mathcal{C}_k$ as the set containing the first k couples in this enumeration.

Now let us prove the (if) part of the theorem. We consider some arbitrary preference profile $\mathcal{P}$ that neither contains a γ-configuration nor a δ-configuration. Our argument is algorithmic in nature. We start from the trivial partition $\mathcal{X}^{(0)}$ of the voters, and then step by step refine this partition while working through $\frac{1}{2}m(m-1)$ phases. The kth such phase generates an ordered partition $\mathcal{X}^{(k)} = \langle X_1^{(k)}, \ldots, X_p^{(k)} \rangle$ of the voters that satisfies the following two properties.

- (i) For $1 \leq j \leq p-1$, the union of parts $X_1^{(k)}, \ldots, X_j^{(k)}$ is separated from the union of parts $X_{j+1}^{(k)}, \ldots, X_p^{(k)}$ by one of the couples in $\mathcal{C}_k$.
- (ii) For every couple in $\mathcal{C}_k$, there is a j with $1 \leq j \leq p-1$ such that the couple separates the union of $X_1^{(k)}, \ldots, X_j^{(k)}$ from the union of $X_{j+1}^{(k)}, \ldots, X_p^{(k)}$.

Note that property (ii) implies that every part $X_j^{(k)}$ is pure with respect to every couple in $\mathcal{C}_k$. The following four lemmas summarize some useful combinatorial observations on the ordered partition $\mathcal{X}^{(k)}$ and how it relates to couple $\{a_{k+1}, b_{k+1}\}$.

Lemma 3.2 *At most one part in the ordered partition $\mathcal{X}^{(k)}$ is mixed with respect to couple $\{a_{k+1}, b_{k+1}\}$.*

Proof. Suppose for the sake of contradiction that the parts $X_s^{(k)}$ and $X_t^{(k)}$ with $1 \leq s < t \leq p$ both are mixed with respect to couple $\{a_{k+1}, b_{k+1}\}$. In other words, part $X_s^{(k)}$ contains a voter V_1' with $a_{k+1} \succ b_{k+1}$ and another voter V_2' with $b_{k+1} \succ a_{k+1}$, and part $X_t^{(k)}$ contains a voter V_3' with $a_{k+1} \succ b_{k+1}$ and another voter V_4' with $b_{k+1} \succ a_{k+1}$.

Property (i) yields the existence of a couple $\{x, y\} \in \mathcal{C}_k$ that separates the union of parts $X_1^{(k)}, \ldots, X_s^{(k)}$ from the union of the parts $X_{s+1}^{(k)}, \ldots, X_p^{(k)}$. In particular, this couple separates $X_s^{(k)}$ from $X_t^{(k)}$. This implies that voters V_1' and V_2' agree on couple $\{x, y\}$ (say, with $x \succ y$), whereas voters V_3' and V_4' have the opposite ranking (say $y \succ x$). Then the four voters V_1', V_2', V_3', and V_4' together with the four alternatives a_{k+1}, b_{k+1}, x, and y form a δ-configuration; this yields the desired contradiction. □

Lemma 3.3 *Consider s and t with $2 \leq s < t \leq p$. If some voter V_1' in part $X_1^{(k)}$ ranks $a_{k+1} \succ b_{k+1}$ and if some voter V_2' in part $X_s^{(k)}$ ranks $b_{k+1} \succ a_{k+1}$, then every voter V_3' in part $X_t^{(k)}$ ranks $b_{k+1} \succ a_{k+1}$.*

Proof. Suppose for the sake of contradiction that the voter V_3' ranks $a_{k+1} \succ b_{k+1}$. Then the couple $\{a_{k+1}, b_{k+1}\}$ separates V_2' from V_1' and V_3'. Property (i) yields a couple $\{x, y\} \in \mathcal{C}_k$ that separates $X_1^{(k)}$ from $X_s^{(k)} \cup X_t^{(k)}$; this couple separates V_1' from V_2' and V_3'. Property (i) yields also a couple $\{u, v\} \in \mathcal{C}_k$ that separates $X_t^{(k)}$ from $X_1^{(k)} \cup X_s^{(k)}$; this couple separates V_3' from V_1' and V_2'.

Then the three voters V_1', V_2', and V_3' together with the six alternatives a_{k+1}, b_{k+1}, x, y, u, and v form a γ-configuration; a contradiction. □

The statement of the following lemma is symmetric to the statement of Lemma 3.3, and it can be proved by symmetric arguments.

Lemma 3.4 *Consider s and t with $1 \leq s < t \leq p-1$. If some voter V_2' in part $X_t^{(k)}$ ranks $a_{k+1} \succ b_{k+1}$ and some voter V_3' in part $X_p^{(k)}$ ranks $b_{k+1} \succ a_{k+1}$, then every voter V_1' in part $X_s^{(k)}$ ranks $a_{k+1} \succ b_{k+1}$.* □

Lemma 3.5 *There exists an index ℓ with $1 \leq \ell \leq p$ such that the couple $\{a_{k+1}, b_{k+1}\}$ separates the union of parts $X_1^{(k)}, \ldots, X_{\ell-1}^{(k)}$ from the union of parts $X_{\ell+1}^{(k)}, \ldots, X_p^{(k)}$.*

Proof. If $p = 1$ or if all voters in the profile agree on the relative ranking of a_{k+1} and b_{k+1}, the choice $\ell = 1$ works. Hence we assume that $p \geq 2$ and that there are two voters who disagree on the ranking of a_{k+1} and b_{k+1}. By Lemma 3.2 the parts $X_1^{(k)}$ and $X_p^{(k)}$ cannot both be mixed with respect to $\{a_{k+1}, b_{k+1}\}$.

If the first part $X_1^{(k)}$ is pure with respect to $\{a_{k+1}, b_{k+1}\}$, we pick an arbitrary voter V_1' from $X_1^{(k)}$. We choose ℓ as the smallest index for which $X_\ell^{(k)}$ contains some voter V_2' who ranks a_{k+1} versus b_{k+1} differently from voter V_1'. Then Lemma 3.3 yields that every voter V_3' in the parts $X_{\ell+1}^{(k)}, \ldots, X_p^{(k)}$ must rank a_{k+1} versus b_{k+1} differently from voter V_1'. Hence the chosen index ℓ has all the desired properties, and this case is closed. In the remaining case the last part $X_p^{(k)}$ is pure with respect to $\{a_{k+1}, b_{k+1}\}$; this case can be settled in a symmetric fashion while using Lemma 3.4. □

Now let us finally describe how to construct the ordered partition $\mathcal{X}^{(k+1)}$ in the $(k+1)$th phase. Our starting point is the ordered partition $\mathcal{X}^{(k)}$, and we determine an index ℓ as defined in Lemma 3.5. If part $X_\ell^{(k)}$ is pure with respect to $\{a_{k+1}, b_{k+1}\}$, then we make the new partition $\mathcal{X}^{(k+1)}$ coincide with the old partition $\mathcal{X}^{(k)}$; properties (i) and (ii) are satisfied in $\mathcal{X}^{(k+1)}$. If part $X_\ell^{(k)}$ is mixed with respect to $\{a_{k+1}, b_{k+1}\}$, then we subdivide it into two parts Y and Z so that $\{a_{k+1}, b_{k+1}\}$ separates the union of parts $X_1^{(k)}, \ldots, X_{\ell-1}^{(k)}, Y$ from the union of parts $Z, X_{\ell+1}^{(k)}, \ldots, X_p^{(k)}$. Then the resulting partition

$$\mathcal{X}^{(k+1)} = \langle X_1^{(k)}, \ldots, X_{\ell-1}^{(k)}, Y, Z, X_{\ell+1}^{(k)}, \ldots, X_p^{(k)} \rangle$$

satisfies properties (i) and (ii) by construction.

We keep working like this and complete phase after phase, until in the very last phase $k = \frac{1}{2}m(m-1)$ we generate the final partition $\mathcal{X}^* = \langle X_1^*, \ldots, X_q^* \rangle$. We construct an ordering π^* of the voters that lists the voters in every part X_j^* before all the voters in part X_{j+1}^* $(1 \leq j \leq q-1)$. Property (ii) guarantees that every couple separates an initial piece of partition $\mathcal{X}^*$ from the complementary final piece, which implies that the ordering π^* for the voters in $\mathcal{P}$ is single-crossing. This completes the proof of Theorem 3.1.

We conclude this section with several comments on the above proof.

(1) Let $\langle X_1^{(k)}, \ldots, X_p^{(k)} \rangle$ be the ordered partition determined in phase k, and consider an ordering σ of the voters that lists the voters in every part $X_j^{(k)}$ before all the voters in the succeeding part $X_{j+1}^{(k)}$. Let ordering σ^- list the voters in reverse order to σ. Then σ and σ^- are single-crossing with respect to all couples in $\mathcal{C}_k$. In fact, *any* ordering that is single-crossing with respect to all couples in $\mathcal{C}_k$ can be constructed in that fashion. This can be established by an inductive argument.

(2) By property (ii), every part X_j^* in the final partition $\mathcal{X}^*$ is pure with respect to every possible couple of alternatives. This means that all voters in part X_j^* have identical preference orderings, and that the ordering π^* is uniquely determined except for swapping voters with identical preference orderings.

(3) The preceding two comments imply the following. Let $\mathcal{P}$ be a preference profile in which distinct voters always have distinct preference orderings. If $\mathcal{P}$ is single-crossing, then there exist exactly two single-crossing orderings of the voters and these two orderings are mirror images of each other.

(4) By property (i), every two consecutive parts X_j^* and X_{j+1}^* must be separated by one of the couples. Since there are only $\frac{1}{2}m(m-1)$ distinct couples, there are at most $\frac{1}{2}m(m-1)+1$ parts in the final partition. This shows that a single-crossing preference profile contains at most $\frac{1}{2}m(m-1)+1$ voters with distinct preference orderings. (This bound of course is already known from the connection between single-crossing profiles and weak Bruhat orders as indicated in Section 2.1.)

4 The size of forbidden configurations

Throughout this short section, we speak of preference profiles with m alternatives and n voters as $m \times n$ configurations. Theorem 3.1 characterizes single-crossing preference profiles through certain forbidden 6×3 and 4×4 configurations. Are there perhaps other characterizations that work with smaller forbidden configurations? The following lemma shows that this is not the case, and hence our characterization uses the smallest possible forbidden configurations.

Lemma 4.1 *Every characterization of single-crossing preference profiles through forbidden configurations must forbid (a) some $m \times n$ configuration with $m \geq 6$ and $n \geq 3$ and (b) some $m \times n$ configuration with $m \geq 4$ and $n \geq 4$.*

Proof. Consider an arbitrary characterization of single-crossing profiles with forbidden configurations $F_1, \ldots, F_k$. Consider the following 6×3 configuration C.

Voter V_1: $b \succ_1 a \succ_1 c \succ_1 d \succ_1 e \succ_1 f$
Voter V_2: $a \succ_2 b \succ_2 d \succ_2 c \succ_2 e \succ_2 f$
Voter V_3: $a \succ_3 b \succ_3 c \succ_3 d \succ_3 f \succ_3 e$

This profile contains a γ-configuration and thus is not single-crossing. If we remove any alternative from C, the resulting 5×3 configuration is single-crossing and cannot be forbidden. And if we remove any voter from C, the resulting 6×2 configuration is again single-crossing and again cannot be forbidden. Hence the only possibility for correctly recognizing C as not single-crossing is by either forbidding C itself or by forbidding appropriate larger configurations that contain C. This proves (a). The proof of (b) is based on the following 4×4 configuration C' which contains a δ-configuration.

Voter V_1: $a \succ_1 b \succ_1 c \succ_1 d$
Voter V_2: $a \succ_2 b \succ_2 d \succ_2 c$
Voter V_3: $b \succ_3 a \succ_3 c \succ_3 d$
Voter V_4: $b \succ_4 a \succ_4 d \succ_4 c$

Since the argument is analogous to the one in (a), we omit the details. □

5 Recognizing the single-crossing property

In this section, we sketch how to produce all (if any) single-crossing orderings of the voters by utilizing the PQ-tree algorithm as developed by Booth & Lueker [7]. The PQ-tree algorithm was designed to recognize, inter alia, *consecutive ones matrices.* A 0-1-matrix has the *consecutive ones property*, if its columns can be permuted such that the ones in each row are consecutive (and hence form an interval).

Hence let us consider an arbitrary preference profile $\mathcal{P}$, and let us transform it into a corresponding 0-1-matrix $M(\mathcal{P})$ in the following way. For each voter, the matrix $M(\mathcal{P})$

contains a corresponding column. For each ordered pair $\langle a, b\rangle$ of alternatives, matrix $M(\mathcal{P})$ has a corresponding row with value 1 at column j if voter j prefers alternative a to alternative b, and value 0 otherwise. For a preference profile with n voters and m alternatives, the resulting 0-1-matrix $M(\mathcal{P})$ has n columns and $m(m-1)$ rows. Example 5.1 illustrates this construction for a concrete profile with four voters and three alternatives.

Example 5.1 (A single-crossing profile and its 0-1-matrix representation) *Suppose that there are four voters V_1, V_2, V_3, and V_4 voting over three alternatives 1, 2, and 3. The preference orderings of the voters are as follows:*

Voter V_1: $3 \succ_1 1 \succ_1 2$
Voter V_2: $2 \succ_2 3 \succ_2 1$
Voter V_3: $2 \succ_3 1 \succ_3 3$
Voter V_4: $3 \succ_4 2 \succ_4 1$

Our construction yields the following 0-1-matrix corresponding to this profile.

	V_1	V_2	V_3	V_4
$\langle 1,2\rangle$	1	0	0	0
$\langle 2,1\rangle$	0	1	1	1
$\langle 1,3\rangle$	0	0	1	0
$\langle 3,1\rangle$	1	1	0	1
$\langle 2,3\rangle$	0	1	1	0
$\langle 3,2\rangle$	1	0	0	1

By applying the PQ-tree algorithm of Booth & Lueker [7], one can find all permutations of the columns with the consecutive ones property. One possible consecutive ones permutation of the columns is $\langle V_1, V_4, V_2, V_3\rangle$. As one can easily verify, this is also a single-crossing ordering of the voters in the original profile. □

Lemma 5.2 *A preference profile $\mathcal{P}$ is single-crossing if and only if the corresponding 0-1-matrix $M(\mathcal{P})$ has the consecutive ones property.*

Proof. An ordering of the voters is single-crossing for $\mathcal{P}$ if and only if this ordering permutes the columns of $M(\mathcal{P})$ so that the ones in each row are consecutive. □

The PQ-algorithm [7] solves the consecutive ones matrix problem in $O(x + y + z)$ time, where x and y are respectively the number of columns and rows, and z is the total number of 1s in the matrix. Hence, single-crossing profiles can be recognized in $O(m^2 + n + nm^2) = O(nm^2)$ time.

6 Conclusion

In this paper, we give an equivalent characterization of single-crossing preferences through two minimal forbidden substructures: γ- and δ-configurations. We demonstrate the close relation between single-crossing preferences and weak Bruhat orders. Futhermore, we can find all single-crossing orderings of a preference profile by transforming them into a binary matrix and asking whether this matrix has the consecutive ones property. This process needs subquadratic time and utilizes the consecutive ones matrix problem. Hence, searching for a direct and more efficient way of detecting the single-crossing property would be an interesting challenge.

References

[1] J. Abello (1991). The weak Bruhat order of S_Σ, consistent sets, and Catalan numbers. *SIAM Journal on Discrete Mathematics 4*, 1–16

[2] M.A. Ballester and G. Haeringer (2011). A characterization of the single-peaked domain. *Social Choice and Welfare 36*, 305–322.

[3] S. Barberà and M.O. Jackson (2004). Choosing how to choose: Self-stable majority rules and constitutions. *Quarterly Journal of Economics 119*, 1011–1048.

[4] S. Barberà and B. Moreno (2011). Top monotonicity: A common root for single peakedness, single crossing and the median voter result. *Games and Economic Behavior 73*, 345–359.

[5] D. Black (1948). On the rationale of group decision-making. *Journal of Political Economics 56*, 23–34.

[6] M. Bóna (2004). *Combinatorics of Permutations*. Chapman and Hall/CRC.

[7] K.S. Booth and G. S. Lueker (1976). Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences 13*, 335–379.

[8] G. Demange (1994). Intermediate preferences and stable coalition structures. *Journal of Mathematical Economics 23*, 45–58.

[9] D. Epple and G.J. Platt (1998). Equilibrium and local redistribution in an urban economy when households differ in both preferences and incomes. *Journal of Urban Economics 43*, 23–51.

[10] A. Galambos and V. Reiner (2008). Acyclic sets of linear orders via the Bruhat orders. *Social Choice and Welfare 30*, 245–264.

[11] J.S. Gans and M. Smart (1996). Majority voting with single-crossing preferences. *Journal of Public Economics 59*, 219–237.

[12] J.-M. Grandmont (1978). Intermediate preferences and majority rule. *Econometrica 46*, 317–330.

[13] A.J. Hoffman, A.W.J. Kolen, and M. Sakarovitch (1985). Totally-balanced and greedy matrices. *SIAM Journal on Algebraic Discrete Methods 6*, 721–730.

[14] K. Inada (1969). The simple majority rule. *Econometrica 37*, 490–506.

[15] F.-C. Kung (2006). An algorithm for stable and equitable coalition structures with public goods. *Journal of Public Economic Theory 8*, 345–355.

[16] K. Kuratowski (1930). Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae 15*, 271–283.

[17] C. Lekkerkerker and D. Boland (1962). Representation of finite graphs by a set of intervals on the real line. *Fundamenta Mathematicae 51*, 45–64.

[18] A.H. Meltzer and S.F. Richard (1981). A rational theory of the size of government. *Journal of Political Economy 89*, 914–927.

[19] H. Moulin (1980). On strategy-proofness and single peakedness. *Public Choice 35*, 437–455.

[20] K.W.S. Roberts (1977). Voting over income tax schedules. *Journal of Public Economics 8*, 329–340.

[21] P. Rothstein (1990). Order restricted preferences and majority rule. *Social Choice and Welfare 7*, 331–342.

[22] A. Saporiti (2009). Strategy-proofness and single-crossing. *Theoretical Economics 4*, 127-163.

[23] A. Saporiti and F. Tohmé (2006). Single-crossing, strategic voting and the median choice rule. *Social Choice and Welfare 26*, 363–383.

[24] F. Westhoff (1977). Existence of equilibria in economies with a local public good. *Journal of Economic Theory 14*, 84–112.

Robert Bredereck
Institut für Eletrotechnik und Theoretische Informatik
Technische Universität Berlin
Ernst Reuter Platz 7, Sekr. TEL 5-1, 10587 Berlin, Germany
Email: `robert.bredereck@tu-berlin.de`

Jiehua Chen
Institut für Eletrotechnik und Theoretische Informatik
Technische Universität Berlin
Ernst Reuter Platz 7, Sekr. TEL 5-1, 10587 Berlin, Germany
Email: `jiehua.chen@tu-berlin.de`

Gerhard J. Woeginger
Department of Mathematics and Computer Science
TU Eindhoven
P.O. Box 513, 5600 MB Eindhoven, the Netherlands
Email: `gwoegi@win.tue.nl`

The Price of Neutrality for the Ranked Pairs Method*

Markus Brill and Felix Fischer

Abstract

The complexity of the winner determination problem has been studied for almost all common voting rules. A notable exception, possibly caused by some confusion regarding its exact definition, is the method of ranked pairs. The original version of the method, due to Tideman, yields a social preference function that is irresolute and neutral. A variant introduced subsequently uses an exogenously given tie-breaking rule and therefore fails neutrality. The latter variant is the one most commonly studied in the area of computational social choice, and it is easy to see that its winner determination problem is computationally tractable. We show that by contrast, computing the set of winners selected by Tideman's original ranked pairs method is NP-complete, thus revealing a trade-off between tractability and neutrality. In addition, several results concerning the hardness of manipulation and the complexity of computing possible and necessary winners are shown to follow as corollaries from our findings.

1 Introduction

The fundamental problem of social choice theory can be concisely described as follows: given a number of individuals, or *voters*, each having a preference ordering over a set of *alternatives*, how can we aggregate these preferences into a collective, or *social*, preference ordering that is in some sense faithful to the individual preferences? By a preference ordering we here understand a (transitive) ranking of all alternatives, and a function aggregating individual preference orderings into social preference orderings is called a *social preference function* (SPF).[1]

A natural idea to construct an SPF is by letting an alternative a be socially preferred to another alternative b if and only if a majority of voters prefers a to b. However, it was observed as early as the 18th century that this approach might lead to paradoxical situations: the collective preference relation may be cyclic even when all individual preferences are transitive [7].

To remedy this situation, a large number of SPFs have been suggested, together with a variety of criteria that a reasonable SPF should satisfy. *Neutrality* and *anonymity*, for instance, are basic fairness criteria which require, loosely speaking, that all alternatives and all voters are treated equally. Another criterion we will be interested in is the *computational effort* required to evaluate an SPF. Computational tractability of the winner determination problem is obviously a significant property of any SPF: the inability to efficiently compute social preferences would render the method virtually useless, at least for large problem instances that do not exhibit additional structure. As a consequence, computational aspects of preference aggregation have received tremendous interest in recent years (see, e.g., [9, 5, 4]).

In this paper, we study the computational complexity of the *ranked pairs* method [15]. To the best of our knowledge, this question has not been considered before, which is particularly

*An earlier version of this paper has appeared in the proceedings of AAAI-2012.

[1] In contrast to a *social welfare function* as studied by Arrow [1], an SPF can output *multiple* social preference orderings with the interpretation that all those rankings are tied for winner. The rationale behind this is to allow for a symmetric outcome when individual preferences are symmetric, like in the case of two individuals with diametrically opposed preferences.

surprising given the extensive literature that is concerned with computational aspects of ranked pairs.[2] A possible reason for this gap might be the confusion of two variants of the method, only one of which satisfies neutrality. In Section 2, we address this confusion and describe both variants. After introducing the necessary notation in Section 3, we show in Section 4 that deciding whether a given alternative is a ranked pairs winner for the neutral variant is NP-complete. Section 5 presents a number of corollaries, and Section 6 discusses variants of the ranked pairs method that are not anonymous.

2 Two Variants of the Ranked Pairs Method

In this section we address the difference between two variants of the ranked pairs method that are commonly studied in the literature. Both variants are anonymous, i.e., treat all voters equally. Non-anonymous variants of the ranked pairs method have been suggested by Tideman [15] and Zavist and Tideman [20], and will be discussed in Section 6.

The ranked pairs method is most easily described as the result of the following procedure. First define a "priority" ordering over the set of all unordered pairs of alternatives by giving priority to pairs with a larger majority margin. Then, construct a ranking of the alternatives by starting with the empty ranking and iteratively considering pairs in order of their priority. When pair $\{a, b\}$ is considered, the ranking is extended by fixing that the majority-preferred alternative precedes the other alternative in the ranking, unless this would create a cycle together with the previously fixed pairs, in which case the opposite precedence between a and b is fixed. Clearly, this procedure is guaranteed to terminate with a complete ranking of all alternatives.

What is missing from the above description is a *tie-breaking rule* for cases where two or more pairwise comparisons have the same support from the voters. This turns out to be a rather intricate issue. In principle, it is possible to employ an arbitrary tie-breaking rule. However, each fixed tie-breaking rule biases the method in favor of some alternative and thereby destroys neutrality.[3] In order to avoid this problem, Tideman [15] originally defined the ranked pairs method to return the set of all those rankings that result from the above procedure for *some* tie-breaking rule.[4] We will henceforth denote this variant by RP.

In a subsequent paper, Zavist and Tideman [20] showed that tie-breaking rules of a certain kind are in fact necessary in order to achieve the property of *independence of clones*, which was the main motivation for introducing the ranked pairs method. While Zavist and Tideman [20] proposed a way to define a tie-breaking rule based on the preferences of a distinguished voter (see Section 6 for details), the variant that is most commonly studied in the literature considers the tie-breaking rule to be exogenously given and fixed for all profiles. This variant of ranked pairs will be denoted by RPT. Whereas RP may output a *set* of rankings, with the interpretation that all the rankings in the set are tied for winner, RPT always outputs a single ranking. In social choice terminology, RP is an *irresolute* SPF, and RPT is a *resolute* one. It is straightforward to see that RP is neutral, i.e., treats all alternatives equally, and that RPT is not. An easy example for the latter statement is the case of two alternatives and two voters who each prefer a different alternative.

Rather than completely ranking all alternatives, it is often sufficient to identify the socially "best" alternatives. This is the purpose of a *social choice function* (SCF). An SCF

[2] Typical problems include the hardness of manipulation [3, 18, 14] and the complexity of computing possible and necessary winners [19, 12].

[3] Neutrality can be maintained if the tie-breaking rule varies with the individual preferences (Section 6).

[4] This definition, sometimes called *parallel universes tie-breaking (PUT)*, can also be used to "neutralize" other voting rules that involve tie-breaking [6]. PUT can be interpreted as a *possible winner* notion: if the ranked pairs method is used with an unknown tie-breaking rule, the PUT version of ranked pairs selects exactly those alternatives that have a chance to be chosen in the actual election.

has the same input as an SPF, but returns alternatives instead of rankings. Each SPF gives rise to a corresponding SCF that returns the top elements of the rankings instead of the rankings themselves, and we will frequently switch between these two settings. Interestingly, deciding whether a given ranking is chosen by an SPF can be considerably easier than deciding whether a given alternative is chosen by the corresponding SCF.

From a computational perspective, RPT is easy: constructing the ranking for a given tie-breaking rule takes time polynomial in the size of the input (see Proposition 1). For RP, however, the picture is different: as the number of tie-breaking rules is exponential, executing the iterative procedure for every single tie-breaking rule is infeasible. Of course, this does not preclude the existence of a clever algorithm that efficiently computes the set of all alternatives that are the top element of some chosen ranking.[5] Our main result states that such an algorithm does not exist unless P equals NP.[6]

3 Preliminaries

For a finite set X, let $\mathcal{L}(X)$ denote the set of all *rankings* of X, where a ranking of X is a complete, transitive, and asymmetric relation on X. The *top element* of a ranking $L \in \mathcal{L}(X)$, denoted by $top(L)$, is the unique element $x \in X$ such that $x\ L\ y$ for all $y \in X \setminus \{x\}$. Furthermore, $\binom{X}{2}$ denotes the set of all two-element subsets of X.

Let $N = \{1, \ldots, n\}$ be a set of voters with preferences over a finite set A of alternatives. The preferences of voter $i \in N$ are represented by a ranking $R_i \in \mathcal{L}(A)$. The interpretation of $a\ R_i\ b$ is that voter i strictly prefers a to b. A *preference profile* is an ordered list containing a ranking for each voter.

A *social choice function* (SCF) f associates with every preference profile R a non-empty set $f(R) \subseteq A$ of alternatives. A *social preference function* (SPF) f associates with every preference profile R a non-empty set $f(R) \subseteq \mathcal{L}(A)$ of rankings of A.

An SCF or SPF is *neutral* if permuting the alternatives in the individual rankings also permutes the set of chosen alternatives, or the set of chosen rankings, in the exact same way. Formally, f is neutral if $f(\pi(R)) = \pi(f(R))$ for all preference profiles R and all permutations π of A. An SCF or SPF is *anonymous* if the set of chosen alternatives, or the set of chosen rankings, does not change when the voters are permuted.

For a given preference profile $R = (R_1, \ldots, R_n)$ and two distinct alternatives $a, b \in A$, the *majority margin* $m_R(a, b)$ is defined as the difference between the number of voters who prefer a to b and the number of voters who prefer b to a, i.e.,

$$m_R(a, b) = |\{i \in N : a\ R_i\ b\}| - |\{i \in N : b\ R_i\ a\}|.$$

Thus, $m_R(b, a) = -m_R(a, b)$ for all distinct $a, b \in A$.

The resolute variant of the ranked pairs method takes as input a preference profile R and a *tie-breaking rule* $\tau \in \mathcal{L}(A \times A)$. It constructs a priority ordering of $\binom{A}{2}$ by ordering all two-element subsets by the size of their majority margin, using τ to break ties: $\{a, b\}$ has priority over $\{c, d\}$ if $|m_R(a, b)| > |m_R(c, d)|$, or if $|m_R(a, b)| = |m_R(c, d)|$ and $(a, b)\ \tau\ (c, d)$.[7] The priority ordering is then used to obtain a ranking $\succ_\tau^R \in \mathcal{L}(A)$ by way of the following iterative procedure. Initialise $\succ_\tau^R$ as the empty relation. Iteratively consider the pair $\{a, b\}$

[5] As the number of chosen *rankings* might be exponential, it immediately follows that computing all of them requires exponential time in the worst case.

[6] A similar discrepancy can be observed for an SCF known as the *Banks set* [2]. Whereas Woeginger [17] has proven that computing Banks winners is NP-complete, Hudry [10] has shown that an arbitrary Banks winner can be found efficiently.

[7] Here we assume without loss of generality that the pairs (a, b) and (c, d) are ordered in such a way that $(a, b)\ \tau\ (b, a)$ and $(c, d)\ \tau\ (d, c)$.

with the highest priority among all pairs in $\binom{A}{2}$ that have not been considered so far. There are two cases.

- *Case 1:* $|m_R(a,b)| \neq 0$. Without loss of generality assume $m_R(a,b) > 0$. If the relation $\succ_\tau^R \cup\{(a,b)\}$ is acyclic, the (ordered) pair (a,b) is added to the relation $\succ_\tau^R$. Otherwise, the pair (b,a) is added to $\succ_\tau^R$.

- *Case 2:* $|m_R(a,b)| = 0$. Without loss of generality assume $(a,b)\ \tau\ (b,a)$. If the relation $\succ_\tau^R \cup\{(a,b)\}$ is acyclic, the pair (a,b) is added to the relation $\succ_\tau^R$. Otherwise, the pair (b,a) is added to $\succ_\tau^R$.

After all pairs in $\binom{A}{2}$ have been considered, $\succ_\tau^R$ is a ranking of A. The resolute variant of ranked pairs, interpreted as an SCF, returns the top element of $\succ_\tau^R$.

Definition 1. $\mathsf{RPT}(R,\tau) = \{top(\succ_\tau^R)\}$.

RPT depends on the choice of τ, and it is not neutral. Tideman [15] defined an irresolute and neutral variant that chooses all alternatives that are at the top of $\succ_\tau^R$ for *some* tie-breaking rule τ.

Definition 2. $\mathsf{RP}(R) = \{a \in A : \textit{there exists } \tau \in \mathcal{L}(A \times A) \textit{ such that } a = top(\succ_\tau^R)\}$.

The alternatives in $\mathsf{RP}(R)$ are called *ranked pairs winners for R*. In the SPF setting, RP returns the rankings $\{\succ_\tau^R : \tau \in \mathcal{L}(A \times A)\}$, which are henceforth called *ranked pairs rankings for R*.

We will work with an alternative characterization of ranked pairs rankings that was introduced by Zavist and Tideman [20]. Given a preference profile R, a ranking L of A, and two alternatives a and b, we say that *a attains b through L* if there exists a sequence of distinct alternatives $a_1, a_2, \dots, a_t$, where $t \geq 2$, such that $a_1 = a$, $a_t = b$, $a_i\ L\ a_{i+1}$, and

$$m_R(a_i, a_{i+1}) \geq m_R(b,a) \text{ for all } i \text{ with } 1 \leq i < t.$$

In this case, we will say that a attains b *via* $(a_1, a_2, \dots, a_t)$. A ranking L is called a *stack* if for any pair of alternatives a and b it holds that $a\ L\ b$ implies that a attains b through L.

Lemma 1 (Zavist and Tideman [20])**.** *A ranking of A is a ranked pairs ranking if and only if it is a stack.*

It follows that an alternative is a ranked pairs winner if and only if it is the top element of a stack.

4 Complexity of Winner Determination

We are now ready to study the computational complexity of RP. We first consider the SPF setting and observe that finding and checking ranked pairs rankings is easy. This also provides an efficient way to find *some* ranked pairs winner, i.e., some alternative that is chosen in the SCF setting. The problem of deciding whether a *particular* alternative is a ranked pairs winner, on the other hand, turns out to be NP-complete. Finally, we extend the hardness result to a variant of the winner determination problem that asks for unique winners.

4.1 Ranked Pairs Rankings

It can easily be seen that an arbitrary ranked pairs ranking can be found efficiently.

Proposition 1. *Finding a ranked pairs ranking is in P.*

Proof. We fix some arbitrary tie-breaking rule $\tau \in \mathcal{L}(A \times A)$ and compute $\succ_\tau^R$, which, by definition, is a ranked pairs ranking. When constructing $\succ_\tau^R$, in each round we have to check whether the addition of a pair (a, b) to the relation $\succ_\tau^R$ creates a cycle. This can efficiently be done with a depth-first search. $\square$

Deciding whether a given ranking is a ranked pairs ranking is also feasible in polynomial time, by checking whether the given ranking is a stack.

Proposition 2. *Deciding whether a given ranking is a ranked pairs ranking is in P.*

Proof. By Lemma 1, it suffices to check whether the given ranking L is a stack. This reduces to checking, for every pair (a, b) with $a \; L \; b$, whether a attains b through L. Let a and b with $a \; L \; b$ be given, and define $w = m_R(b, a)$. We construct a directed graph with vertex set A as follows. For all $x, y \in A$, there is an edge from x to y if and only if $x \; L \; y$ and $m_R(x, y) \geq w$. It is easily verified that a attains b through L if and only if there exists a path from a to b in this graph. The latter property can be efficiently checked with a depth-first search. Since the number of pairs in L is polynomial, this proves the statement. $\square$

4.2 Ranked Pairs Winners

We now consider the SCF setting. As every ranked pairs ranking yields a ranked pairs winner, Proposition 1 immediately implies that an arbitrary element of $\mathsf{RP}(R)$ can be found efficiently.

Corollary 1. *Finding a ranked pairs winner is in P.*

Deciding whether a given alternative is a ranked pairs winner, on the other hand, turns out to be NP-complete.

Theorem 1. *Deciding whether a given alternative is a ranked pairs winner is NP-complete.*

Membership in NP follows from Proposition 2. For hardness, we give a reduction from the NP-complete Boolean satisfiability problem (SAT, see, e.g., [13]). An instance of SAT consists of a Boolean formula $\varphi = C_1 \wedge \cdots \wedge C_k$ in conjunctive normal form over a finite set $V = \{v_1, \ldots, v_m\}$ of variables. Denote by $X = \{v_1, \overline{v}_1, \ldots, v_m, \overline{v}_m\}$ the set of all *literals*, where a literal is either a variable or its negation. Each *clause* C_j is a set of literals. An *assignment* $\alpha \subseteq X$ is a subset of the literals with the interpretation that all literals in α are set to "true." Assignment α is *valid* if $\ell \in \alpha$ implies $\overline{\ell} \notin \alpha$ for all $\ell \in X$, and α *satisfies* clause C_j if $C_j \cap \alpha \neq \emptyset$. A valid assignment that satisfies all clauses of φ is a *satisfying assignment* for φ, and a formula that has a satisfying assignment is called *satisfiable.*

For a particular Boolean formula $\varphi = C_1 \wedge \cdots \wedge C_k$ over a set $V = \{v_1, \ldots, v_m\}$ of variables, we will construct a preference profile R_φ over a set A_φ of alternatives such that a particular alternative $d \in A_\varphi$ is a ranked pairs winner for R_φ if and only if φ is satisfiable.

Let us first define the set A_φ of alternatives. For each variable $v_i \in V$, $1 \leq i \leq m$, there are four alternatives v_i, $\bar{v}_i$, v_i', and $\bar{v}_i'$. For each clause C_j, $1 \leq j \leq k$, there is one alternative y_j. Finally, there is one alternative d for which we want to decide membership in $\mathsf{RP}(R_\varphi)$.

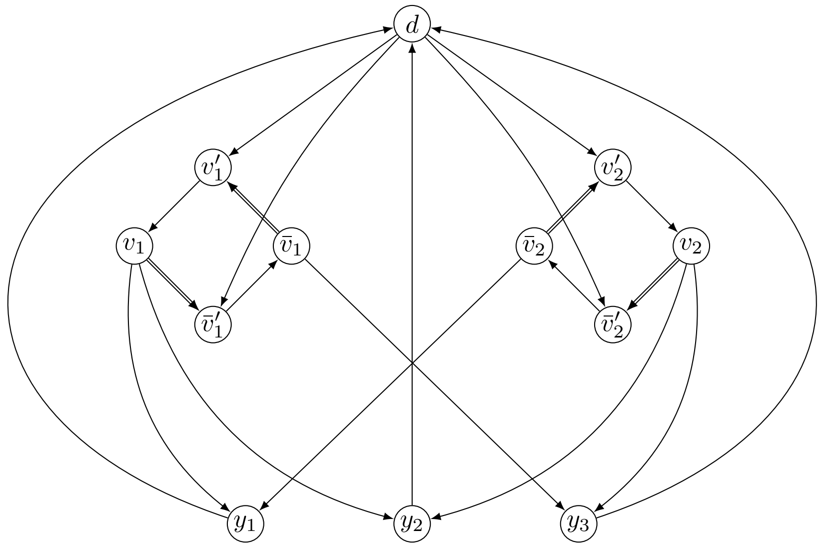


Figure 1: Graphical representation of $m_{R_\varphi}(\cdot,\cdot)$ for the Boolean formula $\varphi = \{v_1, \bar{v}_2\} \wedge \{v_1, v_2\} \wedge \{\bar{v}_1, v_2\}$. The relation $\succ^2$ is represented by arrows, and $\succ^4$ is represented by double-shafted arrows. For all pairs (a, b) that are not connected by an arrow, we have $m(a, b) = m(b, a) = 0$.

Instead of constructing R_φ explicitly, we will specify a number $m(a, b)$ for each pair $(a, b) \in A_\varphi \times A_\varphi$. Debord [8] has shown that there exists a preference profile R such that $m_R(a, b) = m(a, b)$ for all a, b, as long as $m(a, b) = -m(b, a)$ for all a, b and all the numbers $m(a, b)$ have the same parity.[8] In order to conveniently define $m(\cdot,\cdot)$, the following notation will be useful: for a natural number w, $a \succ^w b$ denotes setting $m(a, b) = w$ and $m(b, a) = -w$.

For each variable $v_i \in V$, $1 \leq i \leq m$, let $v_i \succ^4 \bar{v}_i' \succ^2 \bar{v}_i \succ^4 v_i' \succ^2 v_i$. For each clause C_j, $1 \leq j \leq k$, let $v_i \succ^2 y_j$ if variable $v_i \in V$ appears in clause C_j as a positive literal, and $\bar{v}_i \succ^2 y_j$ if variable v_i appears in clause C_j as a negative literal. Finally let $y_j \succ^2 d$ for $1 \leq j \leq k$ and $d \succ^2 v_i'$ and $d \succ^2 \bar{v}_i'$ for $1 \leq i \leq m$. For all pairs (a, b) for which $m(a, b)$ has not been specified so far, let $m(a, b) = m(b, a) = 0$. An example is shown in Figure 1.

As $m(a, b) \in \{-4, -2, 0, 2, 4\}$ for all $a, b \in A_\varphi$, Debord's theorem guarantees the existence of a preference profile R_φ with $m_{R_\varphi}(a, b) = m(a, b)$ for all $a, b \in A_\varphi$, and such a profile can in fact be constructed efficiently, i.e., in polynomial time.

The following two lemmata show that alternative d is a ranked pairs winner for R_φ if and only if the formula φ is satisfiable.

Lemma 2. *If $d \in RP(R_\varphi)$, then φ is satisfiable.*

Proof. Assume that d is a ranked pairs winner for R_φ and let L be a stack with $top(L) = d$. Consider an arbitrary j with $1 \leq j \leq k$. As L is a stack and $d\ L\ y_j$, d attains y_j through L, i.e., there exists a sequence $P_j = (a_1, a_2, \ldots, a_t)$ with $a_1 = d$ and $a_t = y_j$ such that $a_i\ L\ a_{i+1}$ and $m(a_i, a_{i+1}) \geq 2$ for all i with $1 \leq i < t$. If d attains y_j via several sequences, fix one of them arbitrarily.

[8]Also see the article by Le Breton [11].

The definition of $m(\cdot,\cdot)$ implies that

$$P_j = (d, \overline{\ell}', \overline{\ell}, \ell', \ell, y_j) \quad \text{or}$$
$$P_j = (d, \ell', \ell, y_j),$$

where ℓ is some literal. The former is in fact not possible because $m(\ell, \overline{\ell}') = 4$ implies that $\overline{\ell}'$ does *not* attain ℓ through L. Therefore, each P_j is of the form $P_j = (d, \ell', \ell, y_j)$ for some $\ell \in X$.

Now define assignment α as the set of all literals that are contained in one of the sequences $P_j, 1 \le j \le k$, i.e., $\alpha = X \cap (\bigcup_{j=1}^k P_j)$. We claim that α is a satisfying assignment for φ.

In order to show that α is valid, suppose there exists a literal $\ell \in X$ such that both ℓ and $\overline{\ell}$ are contained in α. This implies that there exist i and j such that d attains y_i via $P_i = (d, \ell', \ell, y_i)$ and d attains y_j via $P_j = (d, \overline{\ell}', \overline{\ell}, y_j)$. In particular, $\ell' \; L \; \ell$ and $\overline{\ell}' \; L \; \overline{\ell}$. It follows that either $\ell' \; L \; \overline{\ell}$ or $\overline{\ell}' \; L \; \ell$, as otherwise $(\ell, \overline{\ell}', \overline{\ell}, \ell')$ would form an L-cycle, contradicting the transitivity of L. However, neither does ℓ' attain $\overline{\ell}$ through L, nor does $\overline{\ell}'$ attain ℓ through L, a contradiction.

In order to see that α satisfies φ, consider an arbitrary clause C_j. As d attains y_j via $P_j = (d, \ell', \ell, y_j)$ and $m(y_j, d) = 2$, we have that $m(\ell, y_j) \ge 2$. By definition of $m(\cdot,\cdot)$, this implies that $\ell \in C_j$. □

Lemma 3. *If φ is satisfiable, then $d \in RP(R_\varphi)$.*

Proof. Assume that φ is satisfiable and let α be a satisfying assignment. Let $V_i = \{v_i, \bar{v}_i, v_i', \bar{v}_i'\}$, $1 \le i \le m$, and $Y = \{y_1, y_2, \ldots, y_k\}$. We define a ranking L of A_φ as follows, using $B \; L \; C$ as shorthand for $b \; L \; c$ for all $b \in B$ and $c \in C$.

- For all $1 \le i \le m$, let $d \; L \; V_i$ and $V_i \; L \; Y$.
- For all $1 \le i < j \le m$, let $V_i \; L \; V_j$.
- For the definition of L within V_i, we distinguish two cases. If $v_i \in \alpha$, i.e., if v_i is set to "true" under α, let $\overline{v}_i \; L \; v_i' \; L \; v_i \; L \; \overline{v}_i'$. If, on the other hand, $v_i \notin \alpha$, let $v_i \; L \; \overline{v}_i' \; L \; \overline{v}_i \; L \; v_i'$.
- Within Y, define L arbitrarily.

We now prove that L is a stack. For each pair (a, b) with $a \; L \; b$, we need to verify that a attains b through L. If $m(b, a) \le 0$, it is easily seen that a attains b through L. We can therefore assume that $m(b, a) > 0$. By definition of L and $m(\cdot,\cdot)$, a particular such pair (a, b) satisfies either

$$a = d \text{ and } b \in Y, \text{ or}$$
$$a, b \in V_i \text{ for some } i \text{ with } 1 \le i \le m.$$

First consider a pair of the former type, i.e., $(a, b) = (d, y_j)$ for some j with $1 \le j \le k$. As α satisfies C_j, there exists $\ell \in C_j$ with $\ell \in \alpha$. Consider the sequence $P_j = (d, \ell', \ell, y_j)$. As $m(y_j, d) = 2$ and $d \succ^2 \ell' \succ^2 \ell \succ^2 y_j$, d attains y_j via P_j.

Now consider a pair of the latter type, i.e., $a, b \in V_i$ for some i with $1 \le i \le m$. Assume that $v_i \in \alpha$ and, therefore, $\overline{v}_i \; L \; v_i' \; L \; v_i \; L \; \overline{v}_i'$. The only non-trivial case is the pair $(\overline{v}_i, \overline{v}_i')$ with $\overline{v}_i \; L \; \overline{v}_i'$ and $m(\overline{v}_i', \overline{v}_i) = 2$. But $\overline{v}_i$ attains $\overline{v}_i'$ via $(\overline{v}_i, v_i', v_i, \overline{v}_i')$ because $\overline{v}_i \succ^4 v_i' \succ^2 v_i \succ^4 \overline{v}_i'$. The case $v_i \notin \alpha$ is analogous.

We have shown that L is a stack. Lemma 1 now implies that $d \in \mathsf{RP}(R_\varphi)$, which completes the proof. □

Combining Lemma 2 and Lemma 3, and observing that both A_φ and R_φ can be constructed efficiently, completes the proof of Theorem 1.

4.3 Unique Winners

An interesting variant of the winner determination problem concerns the question whether a given alternative is the *unique* winner for a given preference profile. Despite its similarity to the original winner determination problem, this problem is sometimes considerably easier.[9] For RP, the picture is different: verifying unique winners is not feasible in polynomial time, unless P equals coNP.

Theorem 2. *Deciding whether a given alternative is the unique ranked pairs winner is coNP-complete.*

Proof. Membership in coNP follows from the observation that for every "no" instance there is a stack whose top element is different from the alternative in question.

For hardness, we modify the construction from Section 4.2 to obtain a reduction from the problem UNSAT, which asks whether a given Boolean formula is *not* satisfiable. For a Boolean formula φ, define $A'_\varphi = A_\varphi \cup \{d^*\}$, where d^* is a new alternative and A_φ is defined as in Section 4.2. R'_φ is defined such that $d \succ^2 d^*$ and $d^* \succ^4 a$ for all $a \in A_\varphi \setminus \{d\}$. Within A_φ, R'_φ coincides with R_φ. We show that $\mathsf{RP}(R'_\varphi) = \{d^*\}$ if and only if φ is unsatisfiable.

For the direction from left to right, assume for contradiction that $\mathsf{RP}(R'_\varphi) = \{d^*\}$ and φ is satisfiable. Consider a satisfying assignment α and let L be the ranking of A_φ defined in the proof of Lemma 3. Define the ranking L' of A'_φ by

$$L' = L \cup \{(d, d^*)\} \cup \{(d^*, a) : a \in A_\varphi \setminus \{d\}\}.$$

That is, L' extends L by inserting the new alternative d^* in the second position. As in the proof of Lemma 3, it can be shown that L' is a stack. It follows that $top(L') = d \in \mathsf{RP}(R'_\varphi)$, contradicting the assumption that $\mathsf{RP}(R'_\varphi) = \{d^*\}$.

For the direction from right to left, assume for contradiction that φ is unsatisfiable and $\mathsf{RP}(R'_\varphi) \neq \{d^*\}$. Then there exists a tie-breaking rule τ such that $top(\succ_\tau^{R'_\varphi}) = a \neq d^*$. From the definition of R'_φ it follows that $a = d$, as $d^* \succ^4 b$ for all $b \in A_\varphi \setminus \{d\}$ and there are no $\succ^4$-cycles. By the same argument as in the proof of Lemma 2, it can be shown that φ is satisfiable, contradicting our assumption. □

5 New Proofs for Old and New Results

In this section we briefly consider computational problems other than winner determination. We show that our findings imply several hardness results, some of which are already known. We also point out some errors in the literature that are due to the assumption that winner determination for ranked pairs is in P. By Theorem 1, this assumption is incorrect unless P=NP. All results concern the neutral variant RP, and we refer to the respective papers for formal definitions of the computational problems.

An alternative a is a *possible winner* for a partially specified preference profile R if there exists a completion R' of R such that a is a winner for R'. It is a *necessary winner* if it is a winner for every completion of R. Both the possible and the necessary winner problem have a variant that requires an alternative to be the *unique* winner for the completions.

Corollary 2. *Computing possible ranked pairs winners is NP-complete. Computing possible* unique *ranked pairs winners is both NP-hard and coNP-hard.*

[9]The Banks set, discussed in Footnote 6, constitutes an example: although deciding membership is NP-complete in general, it can be checked in polynomial time whether an alternative is the *unique* Banks winner. The reason for the latter is that an alternative is the unique Banks winner if and only if it is a Condorcet winner.

Proof. NP-completeness of the non-unique variant was already shown by Xia and Conitzer [18]. Membership in NP holds because for every "yes" instance there exists a completion and a tie-breaking rule that yields the alternative in question. Hardness also follows from Theorem 1, because the possible winner problem is equivalent to the winner determination problem in the special case when the preference profile is completely specified.

NP-hardness of the unique variant was shown by Xia and Conitzer [18]; coNP-hardness follows from Theorem 2, because the possible unique winner problem is equivalent to the unique winner determination problem in the special case when the preference profile is completely specified. Xia and Conitzer [18] in fact claimed NP-completeness, but their argument for membership in NP assumes that winner determination is in P. □

Corollary 3. *Computing necessary ranked pairs winners is both NP-hard and coNP-hard. Computing necessary* unique *ranked pairs winners is coNP-complete.*

Proof. Hardness of the non-unique variant for coNP was shown by Xia and Conitzer [18]; NP-hardness follows from Theorem 1, because the necessary winner problem is equivalent to the winner determination problem in the special case when the preference profile is completely specified. Xia and Conitzer [18] in fact claim coNP-completeness, but their argument for membership in coNP assumes that winner determination is in P.

Completeness of the unique variant for coNP was shown by Xia and Conitzer [18]. Membership in coNP holds because for every "no" instance there is a completion and a tie-breaking rule that produces a different winner. Hardness also follows from Theorem 2, because the necessary unique winner problem is equivalent to the unique winner determination problem in the special case when the preference profile is completely specified. □

The *unweighted coalitional manipulation* (UCM) problem asks whether it is possible for a group of voters to cast their votes in such a way that a distinguished alternative becomes a (non-unique or unique) winner.

Corollary 4. *The non-unique UCM problem under ranked pairs is NP-complete. The unique UCM problem under ranked pairs is both NP-hard and coNP-hard.*

Proof. NP-completeness of the non-unique variant was already shown by Xia et al. [19].[10] Membership in NP holds because for every "yes" instance there is a preference profile for the manipulators and a tie-breaking rule that outputs the alternative in question. Hardness also follows from Theorem 1, because the non-unique UCM problem with zero manipulators is equivalent to the winner determination problem.

NP-hardness of the unique variant was shown by Xia et al. [19]; coNP-hardness follows from Theorem 2, because the unique UCM problem with zero manipulators is equivalent to the unique winner determination problem. Xia et al. [19] in fact claimed NP-completeness, but their argument for membership in NP assumes that winner determination is in P. □

6 Non-Anonymous Variants

As mentioned in Section 2, Tideman [15] and Zavist and Tideman [20] suggested ways to use the preferences of a distinguished voter, say, a chairperson, to render the ranked pairs method resolute. There are essentially two ways to achieve this, which differ in the point in time when ties are broken. For the sake of simplicity, we only consider the SCF setting in this section.

[10] The proof of Theorem 4.1 by Xia et al. [19] actually works for both RP and RPT (Xia, personal communication, March 29, 2012).

The *a priori* variant uses the preferences of the chairperson to construct a tie-breaking rule $\tau \in \mathcal{L}(A \times A)$, which is then used to compute $\mathsf{RPT}(\cdot, \tau)$. The *a posteriori* variant first computes $\mathsf{RP}(\cdot)$ and then chooses the alternative from this set that is most preferred by the chairperson. Both variants are neutral: if the alternatives are permuted in each ranking, including the ranking of the chairperson, the tie-breaking rule and thus the chosen alternative will change accordingly.

Whereas the *a priori* variant is a special case of RPT and therefore efficiently computable, the *a posteriori* variant is intractable by the results in Section 4. It follows that neutrality and tractability can be reconciled at the expense of anonymity. By moving to *non-deterministic* SCFs, one can even regain anonymity: choosing the chairperson for the *a priori* variant uniformly at random results in a procedure that is neutral, anonymous, and tractable, for appropriate generalizations of anonymity and neutrality to the case of non-deterministic SCFs. The winner determination problem for the *a posteriori* variant remains intractable when the chairperson is chosen randomly.

7 Conclusion

We have studied the complexity of the ranked pairs method. While *some* ranked pairs winner is easy to find, deciding whether a given alternative is a winner turns out to be NP-complete. If one is interested in ranked pairs rankings, both problems are computationally easy.

From a practical point of view, the ranked pairs method is easier than most other intractable SCFs. The reason is that the *expected* number of ties among two or more pairs is rather small. This is particularly true when the number of voters is large compared to the number of alternatives, which is the case in many realistic settings. It is therefore to be expected that ranked pairs winners are easy to compute *on average* for most reasonable distributions of individual preferences.

Our results reveal a trade-off between neutrality and tractability in the context of the ranked pairs method: while the efficiently computable variant RPT fails neutrality, the neutral variant RP is intractable. A very similar trade-off can be observed for the *single transferable vote* rule [6, 16].

We have finally discussed variants of the ranked pairs method that achieve neutrality at the expense of anonymity, by using individual preferences to break ties. The tractability of those variants depends on the point in time ties are broken.

Acknowledgments This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR 2312/10-1 and FI 1664/1-1. We thank anonymous referees at AAAI and COMSOC for valuable comments and suggestions.

References

[1] K. J. Arrow. *Social Choice and Individual Values.* New Haven: Cowles Foundation, 1st edition, 1951. 2nd edition 1963.

[2] J. S. Banks. Sophisticated voting outcomes and agenda control. *Social Choice and Welfare*, 3:295–306, 1985.

[3] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 53–58, 2009.

[4] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiß, editor, *Multiagent Systems.* MIT Press, 2nd edition, 2012. Forthcoming.

[5] V. Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010.

[6] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 109–115, 2009.

[7] Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.* Imprimerie Royale, 1785. Facsimile published in 1972 by Chelsea Publishing Company, New York.

[8] B. Debord. Caractérisation des matrices des préférences nettes et méthodes d'agrégation associées. *Mathématiques et sciences humaines*, 97:5–17, 1987.

[9] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz.* Springer-Verlag, 2009.

[10] O. Hudry. A note on "Banks winners in tournaments are difficult to recognize" by G. J. Woeginger. *Social Choice and Welfare*, 23:113–114, 2004.

[11] M. Le Breton. On the uniqueness of equilibrium in symmetric two-player zero-sum games with integer payoffs. *Économie publique*, 17(2):187–195, 2005.

[12] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 319–324. AAAI Press, 2011.

[13] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[14] D. Parkes and L. Xia. A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI).* AAAI Press, 2012. Forthcoming.

[15] T. N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[16] B. A. Wichmann. Tie breaking in STV. *Voting Matters*, 19:1–5, 2004.

[17] G. J. Woeginger. Banks winners in tournaments are difficult to recognize. *Social Choice and Welfare*, 20:523–528, 2003.

[18] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011.

[19] L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353. AAAI Press, 2009.

[20] T. M. Zavist and T. N. Tideman. Complete independence of clones in the ranked pairs rule. *Social Choice and Welfare*, 6(2):167–173, 1989.

Markus Brill
Institut für Informatik
Technische Universität München
85748 Garching bei München, Germany
Email: brill@in.tum.de

Felix Fischer
Statistical Laboratory
University of Cambridge
Cambridge CB3 0WB, UK
Email: fischerf@statslab.cam.ac.uk

Group Activity Selection Problem

Andreas Darmann, Edith Elkind, Sascha Kurz, Jérôme Lang, Joachim Schauer,
Gerhard Woeginger

Abstract

We consider a setting where we have to organize one or several group activities for a group of agents. Each agent will participate in at most one activity; her preference over activities generally depends on the number of participants in the activity. The goal is to assign agents to activities in a desirable way. We give a general model, which is a natural generalization of anonymous hedonic games (and can also be expressed, in a less natural way, as a hedonic game). Two well-known solution concepts in hedonic games, namely individual rationality and Nash stability, are particularly meaningful for our model. We study, from the computational point of view, some existence and optimization problems related to these two solution concepts, in the general case as well as for natural restrictions on the agents' preferences.

1 Introduction

There are many real-life situations where a group of agents is faced with a choice of multiple activities, and the members of the group have differing preferences over these activities. Sometimes, it is feasible for the group to split into smaller subgroups, so that each subgroup can pursue its own activity. Consider, for instance, a workshop whose organizers would like to arrange one or more social activities for the free afternoon. The available activities, which will have to take place simultaneously, include a hike, a bus trip, and a table tennis competition. As the activities are scheduled to take place at the same time, each attendee can select at most one activity (or choose not to participate). It is easy enough to elicit the attendees' preferences over the activities, and divide the attendees into groups based on their choices. However, the situation becomes more complicated if one's preferences may depend on the number of other attendees who choose the same activity. For instance, the bus trip has a fixed transportation cost that has to be shared among its participants, which implies that, typically, an attendee i is only willing to go on the bus trip if the number of other participants of the bus trip exceeds a threshold ℓ_i. Similarly, i may only be willing to play table tennis if the number of attendees who signed up for the tournament does *not* exceed a threshold u_i: as there is only one table, the more participants, the less time each individual spends playing.

Neglecting to take the number of participants of each activity into account may lead to highly undesirable outcomes, such as a bus that is shared by two persons, each of them paying a high cost, and a 48-participant table tennis tournament with one table. Adding constraints on the number of participants for each activity is a practical, but imperfect solution, as the agents' preferences over group sizes may differ: while some attendees (say, senior faculty) may be willing to go on the bus trip with just 4–5 other participants, others (say, graduate students) cannot afford it unless the number of participants exceeds 10. A more fine-grained approach is to elicit the agents' preferences over pairs of the form "(activity, number of participants)", rather than over activities themselves, and allocate agents to activities based on this information. In general, agents' preferences can be thought of as weak orders over all such pairs, including the pair "(do nothing, 1)", which we will refer to as the *void activity*. A simpler model, which will be the main focus of this paper, assumes that each agents classifies all pairs into ones that are acceptable to him and ones that are not, and if an agent views his current assignment as unacceptable, he prefers (and is allowed) to switch to the void activity (so the assignment is unstable unless it is acceptable to all agents).

The problem of finding a good assignment of agents to activities, which we will refer to as the *Group Activity Selection Problem* (GASP), may be viewed as a mechanism design problem (or, more narrowly, a voting problem) or as a coalition formation problem, depending on whether we expect the agents to act strategically when reporting their preferences. Arguably, in our motivating example the agents are likely to be honest, so throughout the paper we assume that the central authority knows (or, rather, can reliably elicit) the agents' true preferences, and its goal is to find an assignment of players to activities that, informally speaking, is stable and/or maximizes the overall satisfaction. This model is closely related to that of *anonymous hedonic games* [3], where, just as in our setting, players have to split into groups and each player has preferences over possible group sizes. The main difference between anonymous hedonic games and our problem is that, in our setting, the agents' preferences depend not only on the group size, but also on the activity that has been allocated to their group; thus, our model can be seen as a generalization of anonymous hedonic games. On the other hand, we can represent our problem as a general (i.e., non-anonymous) hedonic game [4, 3], by creating a dummy agent for each activity and endowing it with suitable preferences (see Section 2.2 for details). However, our setting has useful structural properties that distinguish it from a generic hedonic game: for instance, it allows for succinct representation of players' preferences, and, as we will see, has several natural special cases that admit efficient algorithms for finding good outcomes.

In this paper, we initiate the formal study of GASP. Our goal is to put forward a model for this problem that is expressive enough to capture many real-life activity selection scenarios, yet simple enough to admit efficient procedures for finding good assignments of agents to activities. We describe the basic structure of the problem, and discuss plausible constraints of the number and type of available activities and the structure of agents' preferences. We show that even under a fairly simple preference model (where agents are assumed to approve or disapprove each available alternative) finding an assignment that maximizes the number of satisfied agents is computationally hard; however, we identify several natural special cases of the problem that admit efficient algorithms for this problem. We also briefly discuss the issue of stability in our setting.

We do not aim to provide a complete analysis of the group activity selection problem; rather, our work should be seen as a first step towards understanding the algorithmic and incentive issues that arise in this setting. We hope that our paper will lead to future research on this topic; to facilitate this, towards the end of the paper we discuss several possible extensions of our model as well as list some problems left open by our work.

2 Formal Model

Definition 1. *An instance of the* Group Activity Selection Problem (GASP) *is given by a set of* agents $N = \{1, \ldots, n\}$*, a set of* activities $A = A^* \cup \{a_\emptyset\}$*, where* $A^* = \{a_1, \ldots, a_p\}$*, and a* profile P*, which consists of* n votes *(one for each agent):* $P = (V_1, \ldots, V_n)$*. The vote of agent* i *describes his preferences over the set of* alternatives $X = X^* \cup \{a_\emptyset\}$*, where* $X^* = A^* \times \{1, \ldots, n\}$*; alternative* (a, k) *is interpreted as "activity* a *with* k *participants", and* $a_\emptyset$ *is the* void activity.

The vote V_i *of an agent* $i \in N$ *is a weak order over* X^**; for readability we will also denote it by* $\succeq_i$*, and its induced strict preference and indifference relations are denoted by* $\succ_i$ *and* $\sim_i$*, respectively. We set* $S_i = \{(a, k) \in X^* \mid (a, k) \succ_i a_\emptyset\}$*; we say that voter* i approves *of all alternatives in* S_i*, and refer to the set* S_i *as the* induced approval vote *of voter* i*.*

Throughout the paper we will mostly focus on a special case of our problem where no agent is indifferent between the void activity and any non-void activity (i.e., for any $i \in N$ *we have* $\{x \in X \mid x \sim_i a_\emptyset\} = \{a_\emptyset\}$*), and each agent is indifferent between all the alternatives in* S_i*; we denote this special case of our problem by* a-GASP.

It will be convenient to distinguish between activities that are unique and ones that exist in multiple copies. For instance, if there is a single tennis table and two buses, then we can organize one table tennis tournament, two bus trips (we assume that there is only one potential destination for the bus trip, so these trips are identical), and an unlimited number of hikes (again, we assume that there is only one hiking route). This distinction will be useful for the purposes of complexity analysis: for instance, some of the problems we consider are easy when we have k copies of one activity, but hard when we have k distinct activities. Formally, we say that two activities a and b are *equivalent* if for every agent i and every $j \in \{1, \ldots, n\}$ it holds that $(a, j) \sim_i (b, j)$. We say that an activity $a \in A^*$ is *k-copyable* if A^* contains exactly k activities that are equivalent to a (including a itself). We say that a is *simple* if it is 1-copyable; if a is k-copyable for $k \geq n$, we will simply say that it is *copyable* (note that we would never want to organize more than n copies of any activity). If some activities in A^* are equivalent, A^* can be represented more succinctly by listing one representative of each equivalence class, together with the number of available copies. However, as long as we make the reasonable assumption that each activity exists in at most n copies, this representation is at most polynomially more succinct.

Our model can be enriched by specifying a set of *constraints* Γ. One constraint that arises frequently in practice is a *global cardinality* constraint, which specifies a bound K on the number of activities to be organized. More generally, we could also consider more complex constraints on the set of activities that can be organized simultaneously, which can be encoded, e.g., by a propositional formula or a set of linear inequalities. We remark that there can also be external constraints on the number of participants for each activity: for instance, a bus can fit at most 40 people. However, these constraints can be incorporated into agents' preferences, by assuming that all agents view the alternatives that do not satisfy these constraints as unacceptable.

2.1 Special Cases

We now consider some natural restrictions on agents' preferences that may simplify the problem of finding a good assignment. We first need to introduce some additional notation.

Given a vote V_i and an activity $a \in A^*$, let $S_i^{\downarrow a}$ denote the projection of S_i onto $\{a\} \times \{1, \ldots, n\}$. That is, we set

$$S_i^{\downarrow a} = \{k \mid (a, k) \in S_i\}.$$

Example 1. *Let $A^* = \{a, b\}$ and consider an agent i whose vote V_i is given by*

$$(a, 8) \succ_i (a, 7) \succ_i (b, 4) \succ_i (a, 9) \succ_i (b, 3) \succ_i (b, 5) \succ_i (b, 6) \succ_i (a, 6) \succ_i a_\emptyset \succ_i \ldots$$

Then $S_i = \{a\} \times [6, 9] \cup \{b\} \times [3, 6]$ and $S_i^{\downarrow a} = \{6, 7, 8, 9\}$.

We are now ready to define two types of restricted preferences for a-GASP that are directly motivated by our running example, namely, *increasing* and *decreasing* preferences. Informally, under increasing preferences an agent prefers to share each activity with as many other participants as possible (e.g., because each activity has an associated cost, which has to be split among the participants), and under decreasing preferences an agent prefers to share each activity with as few other participants as possible (e.g., because each activity involves sharing a limited resource). Of course, an agent's preferences may also be increasing with respect to some activities and decreasing with respect to others, depending on the nature of each activity. We provide a formal definition for a-GASP only; however, it can be extended to GASP in a straightforward way.

Definition 2. *Consider an instance (N, A, P) of* a-GASP. *We say that the preferences of agent i are* increasing (INC) *with respect to an activity $a \in A^*$ if there exists an integer threshold $\ell_i^a \in \{1, \ldots, n+1\}$ such that $S_i^{\downarrow a} = [\ell_i^a, n]$ (where we assume that $[n+1, n] = \emptyset$).*

Similarly, we say that the preferences of agent i are decreasing (DEC) *with respect to an activity $a \in A^*$ if there exists an integer threshold $u_i^a \in \{0, \ldots, n\}$ such that $S_i^{\downarrow a} = [1, u_i^a]$ (where we assume that $[1, 0] = \emptyset$).*

We say that an instance (N, A, P) of a-GASP *is* increasing *(respectively,* decreasing*) if the preferences of each agent $i \in N$ are increasing (respectively, decreasing) with respect to each activity $a \in A^*$. We say that an instance (N, A, P) of* a-GASP *is* mixed increasing-decreasing (MIX) *if there exists a set $A^+ \subseteq A^*$ such that for each agent $i \in N$ his preferences are increasing with respect to each $a \in A^+$ and decreasing with respect to each $a \in A^- = A^* \setminus A^+$.*

For some activities, an agent may have both a lower and an upper bound for the acceptable group size: e.g., one may prefer to go on a hike with at least 3 other people, but does not want the group to be too large in order to maintain a good pace. In this case, we say that an agent has *interval* (INV) preferences; note that increasing/decreasing/mixed increasing-decreasing preferences are a special case of interval preferences.

Definition 3. *Consider an instance* (N, A, P) *of* a-GASP. *We say that the preferences of agent* i *are* interval (INV) *if for every* $a \in A^*$ *there exists a pair of integer thresholds* $\ell_i^a, u_i^a \in \{1, \ldots, n\}$ *such that* $S_i^{\downarrow a} = [\ell_i^a, u_i^a]$ *(where we assume that* $[i, j] = \emptyset$ *for* $i > j$*).*

Other natural constraints on preferences include restricting the size of S_i (or, more liberally, that of $S_i^{\downarrow a}$ for all $a \in A^*$), or requiring agents to have similar preferences: for instance, one could limit the number of agent *types*, i.e., require that the set of agents can be split into a small number of groups so that the agents in each group have identical preferences. We will not define such constraints formally, but we will indicate if they are satisfied by the instances constructed in the hardness proofs in Section 4.1.

2.2 GASP and Hedonic Games

Recall that a *hedonic game* is given by a set of agents N, and, for each agent $i \in N$, a weak order $\geq_i$ over all coalitions (i.e., subsets of N) that include him. That is, in a hedonic game each agent has preferences over coalitions that he can be a part of. A coalition S, $i \in S$, is said to be *unacceptable* for player i if $\{i\} \geq_i S$, i.e., i prefers being alone to being in S. A hedonic game is said to be *anonymous* if each agent is indifferent among all coalitions of the same size that include him, i.e., for every $i \in N$ and every $S, T \subseteq N \setminus \{i\}$ such that $|S| = |T|$ it holds that $S \cup \{i\} \geq_i T \cup \{i\}$ and $T \cup \{i\} \geq_i S \cup \{i\}$.

At a first glance, it may seem that the GASP formalism is more general than that of hedonic games, since in GASP the agents care not only about their coalition, but also about the activity they have been assigned to. However, we will now argue that GASP can be embedded into the hedonic games framework.

Given an instance of the GASP problem (N, A, P) with $|N| = n$, where the i-th agent's preferences are given by a weak order $\succeq_i$, we construct a hedonic game $H(N, A, P)$ as follows. We create $n + p$ players; the first n players correspond to agents in N, and the last p players correspond to activities in A^*. The last p players are indifferent among all coalitions. For each $i = 1, \ldots, n$, player i ranks every non-singleton coalition with no activity players as unacceptable; similarly, all coalitions with two or more activity players are ranked as unacceptable. The preferences over coalitions with exactly one activity player are derived naturally from the votes: if S, T are two coalitions involving player i, x is the unique activity player in S, and y is the unique activity player in T, then i weakly prefers S to T in $H(N, A, P)$ if and only if $(x, |S| - 1) \succeq_i (y, |T| - 1)$, and i weakly prefers S to $\{i\}$ in $H(N, A, P)$ if and only if $(x, |S| - 1) \succeq_i a_\emptyset$. We emphasize that the resulting hedonic games are not anonymous. Further, while this embedding allows us to apply the standard solution concepts for hedonic games without redefining them, the intuition behind these solution concepts is not always preserved (e.g., because activity players never want to deviate). Therefore, in Section 3, we will provide formal definitions of the relevant hedonic games solution concepts adapted to the setting of a-GASP.

We remark that when A^* consists of a single copyable activity (i.e., there are n activities in A^*, all of them equivalent to each other), GASP become equivalent to anonymous hedonic games. Such games have been studied in detail by Ballester [2], who provides a number of complexity results for them. In particular, he shows that finding an outcome that is core stable, Nash stable or individually stable (see Section 3 for the definitions of some of these concepts in the context of a-GASP) is NP-hard. Clearly, all these complexity results also hold for GASP. However, they do not directly imply similar hardness results for a-GASP.

3 Solution Concepts

Having discussed the basic model of GASP, as well as a few of its extensions and special cases, we are ready to define what constitutes a solution to this problem.

Definition 4. *An* assignment *for an instance (N, A, P) of* GASP *is a mapping $\pi : N \to A$; $\pi(i) = a_\emptyset$ means that agent i does not participate in any activity. Each assignment naturally partitions the agents into at most $|A|$ groups: we set $\pi^0 = \{i \mid \pi(i) = a_\emptyset\}$ and $\pi^j = \{i \mid \pi(i) = a_j\}$ for $j = 1, \ldots, p$. For each $j = 1, \ldots, p$, the agents in π^j form a coalition; also, each agent in π^0 forms a singleton coalition.*

Clearly, not all assignments are equally desirable. As a minimum requirement, no agent should be assigned to a coalition that he deems unacceptable. More generally, we prefer an assignment to be stable, i.e., no agent (or group of agents) should have an incentive to change its activity. Thus, we will now define several *solution concepts*, i.e., classes of desirable assignments. We will state our definitions for a-GASP only, though all of them can be extended to the more general case of GASP in a natural way. Given the connection to hedonic games pointed out in Section 2.2, we will proceed by adapting the standard hedonic game solution concepts to our setting; however, this has to be done carefully to preserve intuition that is specific to our setting.

The first solution concept that we will consider is *individual rationality*.

Definition 5. *Given an instance (N, A, P) of* a-GASP *an assignment $\pi : N \to A$ is said to be* individually rational *if for every agent $i \in N$ such that $\pi(i) = a_j \neq a_\emptyset$ it holds that $(a_j, |\pi^j|) \in S_i$.*

Clearly, if an assignment is not individually rational, there exists an agent that can benefit from abandoning his coalition in favor of the void activity. Further, an individually rational assignment always exists: for instance, we can set $\pi(i) = a_\emptyset$ for all $i \in N$. However, a benevolent central authority would usually want to maximize the number of agents that are assigned to non-void activities. Formally, let $\#(\pi) = |\{i \mid \pi(i) \neq a_\emptyset\}|$ denote the number of agents assigned to a non-void activity. We say that π is *maximum individually rational* if π is individually rational and $\#(\pi) \geq \#(\pi')$ for every individually rational assignment π. Further, we say that π is *perfect* if $\#(\pi) = n$[1]. We denote the size of a maximum individually rational assignment for an instance (N, A, P) by $\#(N, A, P)$. In Section 4, we study the complexity of computing a perfect or maximum individually rational assignment for a-GASP, both for the general model and for the special cases considered in Section 2.1.

Besides individual rationality, there is a number of solution concepts for hedonic games that aim to capture stability against individual or group deviations, such as Nash stability, individual stability, contractual individual stability, and (weak and strong) core stability (see, e.g., [6]). In what follows, due to lack of space, we only provide the formal definition (and some results) for Nash stability. We briefly discuss how to adapt other notions of stability to our setting, but we leave the detailed study of their algorithmic properties as a topic for future work.

Definition 6. *Given an instance (N, A, P) of* a-GASP*, an assignment $\pi : N \to A$ is said to be* Nash stable *if it is individually rational and for every agent $i \in N$ such that $\pi(i) = a_\emptyset$ and every $a_j \in A^*$ it holds that $(a_j, |\pi^j| + 1) \notin S_i$.*

If π is not Nash stable, then there is an agent assigned to a void activity who wants to join a group that is engaged in a non-void activity, i.e., he would have approved of the size of this group and its activity choice if he was one of them. Note that a perfect assignment is Nash stable. The reader can easily verify that our definition is a direct adaptation of the notion of Nash stability in hedonic games: if an assignment is individually rational, the only agents who can profitably deviate are the ones assigned to the void activity.

[1]The terminological similarity with the notion of perfect partition in a hedonic game [1] is not a coincidence; there a perfect partition assigns each agent to her preferred coalition; here a perfect assignment assigns each agent to one of her equally preferred alternatives.

The requirement of Nash stability is considerably stronger than that of individual rationality, and, in general, there are cases where a Nash stable assignment does not exist.

Proposition 1. *For each* $n \geq 2$*, there exists an instance* (N, A, P)*,* $|N| = n$*, of* a-GASP *that does not admit a Nash stable assignment. This holds even if* $|A^*| = 1$ *and all agents have interval preferences.*

Proof. Consider an instance (N, A, P) of a-GASP with $A^* = \{a\}$ and induced approval votes given by $S_1 = \{(a, 1)\}$, $S_2 = \{(a, 2)\}$ and $S_i = \emptyset$ for all $i \geq 3$; note that all approved sets are intervals. Whichever assignment π is chosen, either π is not individually rational or agent 2 wants to join a. ■

In Definition 6 an agent is allowed to join a coalition even if the members of this coalition are opposed to this. In contrast, the notion of *individual stability* only allows a player to join a group if none of the existing group members objects. We remark that if all agents have increasing preferences, individual stability is equivalent to Nash stability: no group of players would object to having new members join.

A related hedonic games solution concept is *contractual individual stability*: under this concept, an agent is only allowed to move from one coalition to another if neither the members of his new coalition nor the members of his old coalition object to the move. However, for a-GASP contractual individual stability is equivalent to individual stability. Indeed, in our model no agent assigned to a non-void activity has an incentive to deviate, so we only need to consider deviations from singleton coalitions.

The solution concepts discussed so far deal with individual deviations; resistance to group deviations is captured by the notion of the *core*. One typically distinguishes between *strong group deviations*, which are beneficial for each member of the deviating group, and *weak* group deviations, where the deviation should be beneficial for at least one member of the deviating group and non-harmful for others; these notions of deviation correspond to, respectively, *weak* and *strong* core. We note that in the context of a-GASP strong group deviations amount to players in π^0 forming a coalition in order to engage in a non-void activity. This observation immediately implies that every instance of a-GASP has a non-empty weak core, and an outcome in the weak core can be constructed by a natural greedy algorithm; we omit the details due to space constraints.

4 Computing Good Outcomes

In this section, we consider the computational complexity of finding a "good" assignment for a-GASP. We mostly focus on finding perfect or maximum individually rational assignment; towards the end of the section, we also consider Nash stability. Besides the general case of our problem, we consider special cases obtained by combining constraints on the number and type of activities (e.g., unlimited number of simple activities, a constant number of copyable activities, etc.) and constraints on voters' preferences (INC, DEC, INV, etc.). Note that if we can find a maximum individually rational assignment, we can easily check if a perfect assignment exists, by looking at the size of our maximum individually rational assignment. Thus, we will state our hardness results for the "easier" perfect assignment problem and phrase our polynomial-time algorithms in terms of the "harder" problem of finding a maximum individually rational assignment.

4.1 Individual Rationality: Hardness Results

We start by presenting four NP-complete results, which show that finding a perfect assignment is hard even under fairly strong constraints on preferences and activities. We remark that this problem is obviously in NP, so in what follows we will only provide the hardness proofs. We omit most proofs in this section due to space constraints.

Our first hardness result applies when we have an unlimited number of simple activities, and the agents' preferences are increasing.

Theorem 1. *It is* NP*-complete to decide whether* a-GASP *admits a perfect assignment, even when all activities in* A^* *are simple and all agents have increasing preferences.*

Proof sketch. We provide a reduction from EXACT COVER BY 3-SETS (X3C). Recall that an instance of X3C is a pair $\langle X, \mathcal{Y}\rangle$, where $X = \{1, \ldots, 3q\}$ and $\mathcal{Y} = \{Y_1, \ldots, Y_p\}$ is a collection of 3-element subsets of $\mathcal{Y}$; it is a "yes"-instance if X can be covered by exactly q sets from $\mathcal{Y}$, and a "no"-instance otherwise. Given an instance $\langle X, \mathcal{Y}\rangle$ of X3C, we construct an instance of a-GASP as follows. We set $N = \{1, \ldots, 3q\}$ and $A^* = \{a_1, \ldots, a_p\}$. For each agent i, we define his vote V_i so that the induced approval vote S_i is given by $S_i = \{(a_j, k) \mid i \in Y_j, k \geq 3\}$, and let $P = (V_1, \ldots, V_n)$. Clearly, (N, A, P) is an instance of a-GASP with increasing preferences. It is not hard to check that $\langle X, \mathcal{Y}\rangle$ is a "yes"-instance of X3C if and only if (N, A, P) admits a perfect assignment. ■

Our second hardness result applies to simple activities and decreasing preferences, and holds even if each agent is willing to share each activity with at most one other agent.

Theorem 2. *It is* NP*-complete to decide whether* a-GASP *admits a perfect assignment, even when all activities in* A^* *are simple, all agents have decreasing preferences, and, moreover, for every agent* $i \in N$ *and every alternative* $a \in A^*$ *we have* $S_i^{\downarrow a} \subseteq \{1, 2\}$.

Proof sketch. Consider the following restricted variant of the problem of scheduling on unrelated machines. There are n jobs and p machines. An instance of the problem is given by a collection of numbers $\{p_{ij} \mid i = 1, \ldots, n, j = 1, \ldots, p\}$, where p_{ij} is the running time of job i on machine j, and $p_{ij} \in \{1, 2, +\infty\}$ for every $i = 1, \ldots, n$ and every $j = 1, \ldots, p$. It is a "yes"-instance if there is a mapping $\rho : \{1, \ldots, n\} \to \{1, \ldots, p\}$ assigning jobs to machines so that the makespan is at most 2, i.e., for each $j = 1, \ldots, p$ it holds that $\sum_{i:\rho(i)=j} p_{ij} \leq 2$. This problem is known to be NP-hard (see the proof of Theorem 5 in [7]).

Given an instance $\{p_{ij} \mid i = 1, \ldots, n, j = 1, \ldots, p\}$ of this problem, we construct an instance of a-GASP as follows. We set $N = \{1, \ldots, n\}$, $A^* = \{a_1, \ldots, a_p\}$. Further, for each agent $i \in N$ we construct a vote V_i so that the induced approval vote S_i satisfies $S_i^{\downarrow a_j} = \{1\}$ if $p_{ij} = 2$, $S_i^{\downarrow a_j} = \{1, 2\}$ if $p_{ij} = 1$, and $S_i^{\downarrow a_j} = \emptyset$ if $p_{ij} = +\infty$. Clearly, these preferences satisfy the constraints in the statement of the theorem, and it can be shown that a perfect assignment for (N, A, P) corresponds to a schedule with makespan of at most 2, and vice versa. ■

Our third hardness result also concerns simple activities in decreasing preferences. However, unlike Theorem 2, it holds even if each agent approves of at most 3 activities. The proof proceeds by a reduction from MONOTONE 3-SAT.

Theorem 3. *It is* NP*-complete to decide whether* a-GASP *admits a perfect assignment, even when all activities in* A^* *are simple, all agents have decreasing preferences, and, moreover, for every agent* $i \in N$ *it holds that* $|\{a \mid S_i^{\downarrow a} \neq \emptyset\}| \leq 3$.

Our fourth hardness result applies even when there is only one copyable activity and every agent approves at most two alternatives; however, the agents' preferences constructed in our proof do not satisfy any of the structural constraints defined in Section 2.1. The proof proceeds by a reduction from X3C.

Theorem 4. *It is* NP*-complete to decide whether* a-GASP *admits a perfect assignment, even when all activities in* A^* *are equivalent (i.e.,* A^* *consists of a single copyable activity* a*) and for every* $i \in N$ *we have* $S_i = \{a\} \times \{x_i, y_i\}$*, where* $\{x_i, y_i\} \subset \{1, 2, \ldots, n\}$.

4.2 Individual Rationality: Easiness Results

The hardness results in Section 4.1 imply that if A^* contains an unbounded number of distinct activities, finding a maximum individually rational assignment is computationally hard, even under strong restrictions on agents' preferences (such as INC or DEC). Thus, we can only hope to develop an efficient algorithm for this problem if we assume that the total number of activities is small (i.e., bounded by a constant), or that most of the activities are equivalent to each other (i.e., there is a small number of copyable activities) and the agents' preferences satisfy additional constraints. We will now consider both of these settings, starting with the case where the size of A^* is bounded by a constant.

Theorem 5. *There exist an algorithm that given an instance of* a-GASP *finds a maximum individually rational assignment and runs in time* $n^{|A^*|}\text{poly}(n)$.

Proof. We will check, for each $r = 0, \ldots, n$, if there is an individually rational assignment π with $\#(\pi) = r$, and output the maximum value of r for which this is the case.

Fix an $r \in \{0, \ldots, n\}$ and let $K = |A^*|$. For every vector $(n_1, \ldots, n_K) \in \{0, \ldots, n\}^K$ that satisfies $n_1 + \cdots + n_K = r$ we will check if there exists an assignment of agents to activities such that for each $j = 1, \ldots, K$ exactly n_j agents are assigned to activity a_j (with the remaining agents being assigned to the void activity), and each agent approves of the resulting assignment. Each check will take poly(n) steps, and there are at most $(n+1)^K$ vectors to be checked; this implies our bound on the running time of our algorithm.

For a fixed vector $(n_1, \ldots, n_K)$, we construct an instance of the network flow problem as follows. Our network has a source s, a sink t, a node i for each player $i = 1, \ldots, n$, and a node a_j for each $a_j \in A^*$. There is an arc of unit capacity from s to each agent, and an arc of capacity n_j from node a_j to the sink. Further, there is an arc of unit capacity from i to a_j if and only if $(a_j, n_j) \in S_i$. It is not hard to see that an integral flow F of size r in this network corresponds to an individually rational assignment of size r. ■

We remark that when A^* consists of a single simple activity a, a maximum individually rational assignment can be found by a simple greedy algorithm.

Proposition 2. *Given an instance* (N, A, P) *of* a-GASP *with* $A^* = \{a\}$*, we can find a maximum individually rational assignment for* (N, A, P) *in time* $O(S \log S)$*, where* $S = \sum_{i \in N} |S_i|$.

Proof. Observe that (N, A, P) admits an individually rational assignment π with $\#(\pi) = k$ if and only if $|\{i \mid (a, k) \in S_i\}| \geq k$. Let $\mathcal{R} = \{(i, k) \mid (a, k) \in S_i\}$; note that $|\mathcal{R}| = S$. We can sort the elements of $\mathcal{R}$ in descending order with respect to their second coordinate in time $O(S \log S)$. Now we can scan $\mathcal{R}$ left to right in order to find the largest value of k such that $\mathcal{R}$ contains at least k pairs that have k as their second coordinate; this requires a single pass through the sorted list. ■

Now, suppose that A^* contains many activities, but most of them are equivalent to each other; for instance, A^* may consist of a single k-copyable activity, for a large value of k. Then the algorithm described in the proof of Theorem 5 is no longer efficient, but this setting still appears to be more tractable than the one with many distinct activities. Of course, by Theorem 4, in the absence of any restrictions on the agents' preferences, finding a maximum individually rational assignment is hard even for a single copyable activity. However, we will now show that this problem becomes easy if we additionally assume that the agents' preferences are increasing or decreasing.

Observe first that for increasing preferences having multiple copies of the same activity is not useful: if there is an individually rational assignment where agents are assigned to multiple copies of an activity, we can reassign these agents to the same copy of this activity without violating individual rationality. Thus, we obtain the following easy corollary to Theorem 5.

Corollary 1. *Let* (N, A, P) *be an instance of* a-GASP *with increasing preferences where* A^* *contains at most* K *activities that are not pairwise equivalent. Then we can find a maximum individually rational assignment for* (N, A, P) *in time* $n^K \text{poly}(n)$*.*

For decreasing preferences, we can simply eliminate all copyable activities. Indeed, consider an instance (N, A, P) of a-GASP where some activity $a \in A^*$ is copyable. Then we can assign each agent $i \in N$ such that $(a, 1) \in S_i$ to his own copy of a; clearly, this will only simplify the problem of assigning the remaining agents to the activities.

It remains to consider the case where the agents' preferences are decreasing, there is a bounded number of copies of each activity, and the number of distinct activities are small. While we do not have a complete solution for this case, we can show that in the case of a single k-copyable activity a natural greedy algorithm succeeds in finding a maximum individually rational assignment.

Theorem 6. *Given a decreasing instance* (N, A, P) *of* a-GASP *where* A^* *consists of a single* k*-copyable activity (i.e.,* $A^* = \{a_1, \ldots, a_k\}$*, and all activities in* A^* *are pairwise equivalent), we can find a maximum individually rational assignment in time* $O(n \log n)$*.*

Proof. Since all activities in A^* are pairwise equivalent, we can associate each agent $i \in N$ with a single number $u_i \in \{0, \ldots, n\}$, which is the maximum size of a coalition assigned to a non-void activity that he is willing to be a part of. We will show that our problem can be solved by a simple greedy algorithm. Specifically, we sort the agents in non-increasing order of u_is. From now on, we will assume without loss of generality that $u_1 \geq \cdots \geq u_n$. To form the first group, we find the largest value of i such that $u_i \geq i$, and assign agents $1, \ldots, i$ to the first copy of the activity. In other words, we continue adding agents to the group as long as the agents are happy to join. We repeat this procedure with the remaining agents until either k groups have been formed or all agents have been assigned to one of the groups, whichever happens earlier.

Clearly, the sorting step is the bottleneck of this procedure, so the running time of our algorithm is $O(n \log n)$. It remains to argue that it produces a maximum individually rational assignment. To show this, we start with an arbitrary maximum individually rational assignment π and transform it into the one produced by our algorithm without lowering the number of agents that have been assigned to a non-void activity. We will assume without loss of generality that π assigns all k copies of the activity (even though this is is not necessarily the case for the greedy algorithm).

First, suppose that $\pi(i) = a_\emptyset$, $\pi(j) = a_\ell$ for some $i < j$ and some $\ell \in \{1, \ldots, k\}$. Then we can modify π by setting $\pi(i) = a_\ell$, $\pi(j) = a_\emptyset$. Since $i < j$ implies $u_i \geq u_j$, the modified assignment is individually rational. By applying this operation repeatedly, we can assume that the set of agents assigned to a non-void activity forms a contiguous prefix of $1, \ldots, n$.

Next, we will argue that for each $\ell = 1, \ldots, k$ the group of agents that are assigned to a_ℓ forms a contiguous subsequence of $1, \ldots, n$. To this end, let us sort the coalitions in π in non-decreasing order according to the smallest value of u_i among the coalition members, breaking ties arbitrarily. That is, we reassign the k copies of our activity to coalitions in π so that $\ell < r$ implies $\min_{i \in \pi^\ell} u_i \leq \min_{i \in \pi^r} u_i$. Now, consider the first coalition π^ℓ in our ordering that is not contiguous, i.e., there exist players x, y, z with $x > y > z$ such that $\pi(x) = \pi(z) = a_\ell$, but $\pi(y) \neq a_\ell$. Note that since the set of agents assigned to a non-void activity forms a contiguous prefix of $1, \ldots, n$, $x > y$, and $\pi(x) = a_\ell$, it follows that $\pi(y) \neq a_\emptyset$. Suppose that $\pi(y) = a_r$; our choice of ℓ implies that $r > \ell$. Now, let us modify π by setting $\pi(y) = a_\ell$, $\pi(z) = a_r$. We claim that the resulting assignment remains individually rational. Indeed, since the original assignment was individually rational, and, in particular, x was happy, and $x > y$ (and hence $|\pi^\ell| \leq u_x \leq u_y$), agent y is happy with the modified assignment. Now, consider agent z. He moved from π^ℓ to π^r. Since the original assignment was individually rational, and, in particular, y was happy, we have $|\pi^r| \leq u_y$. Since $y > z$ implies $u_y \leq u_z$, it follows that z is happy with the modified assignment. By repeatedly applying such swaps, we can ensure that each coalition in π forms a contiguous subsequence of $1, \ldots, n$.

Finally, let us renumber the coalitions in π again, this time according to their first element, i.e., assume that $\ell < r$ implies $\min_{i \in \pi^\ell} i < \min_{i \in \pi^r} i$ (note that this numbering is different from the numbering used in the previous step). Consider the smallest value of ℓ such that π^ℓ differs from the ℓ-th coalition constructed by the greedy algorithm (let us denote it by γ^ℓ), and let i be the first agent in $\pi^{\ell+1}$. The description of the greedy algorithm implies that π^ℓ is a strict subset of γ^ℓ and agent i belongs to γ^ℓ. Thus, if we modify π by moving agent i to π^ℓ, the resulting allocation remains individually rational (since i is happy in γ^ℓ). By repeating this step, we will gradually transform π into the output of the greedy algorithm (possibly discarding some copies of the activity). This completes the proof. ■

The algorithm described in the proof of Theorem 6 can be extended to the case where we have one k-copyable activity a and one simple activity b, and the agents have decreasing preferences over both activities. For each $s = 1, \ldots, n$ we will look for the best solution in which s players are assigned to b; we will then pick the best of these n solutions. For a fixed s let $N_s = \{i \in N \mid (b, s) \in S_i\}$. If $|N_s| < s$, no solution for this value of s exists. Otherwise, we can assign any subset of N_s of size s to b. It is not hard to see that we should simply pick the agents in N_s that have the lowest level of tolerance for a i.e., we order the agents in N_s by the values of u_i^a from the smallest to the largest, and pick the first s agents. Indeed, any assignment that is not of this form can be transformed into one of this form by swaps without breaking the individual rationality constraints. It would be interesting to see if this idea can be extended to the case where instead of a single simple activity b we have a constant number of simple activities or a single k'-copyable activity.

We conclude this section by describing an $O(\sqrt{n})$-approximation algorithm for finding a maximum individually rational assignment in a-GASP with a single copyable activity.

Theorem 7. *There exists a polynomial-time algorithm that given an instance (N, A, P) of* a-GASP *where A^* consists of a single copyable activity a, outputs an individually rational assignment π with $\#(\pi) = \Theta(\frac{1}{\sqrt{n}})\#(N, A, P)$.*

Proof. We will say that an agent i is *active* in π if $\pi(i) \neq a_\emptyset$; we say that a a coalition of agents is *active* if it is assigned to a single copy of a. We construct an individually rational assignment π iteratively, starting from the assignment where no agent is active. Let $N^* = \{i \mid \pi(i) = a_\emptyset\}$ be the current set of inactive agents (initially, we set $N^* = N$). At each step, we find the largest coalition of agents that can be assigned to a single copy of a without breaking the individual rationality constraints, and append this assignment to π. We repeat this step until the inactive agents cannot form another coalition.

Now we compare the number of active agents in π with the number of active agents in an optimal individually rational assignment π^*. To this end, let us denote the active coalitions of π by $B_1, \ldots, B_s$, where $|B_1| \geq \ldots \geq |B_s|$. If $|B_1| \geq \sqrt{n}$, we are done, so assume that this is not the case. Note that since B_1 was chosen greedily, this implies that $|C| \leq \sqrt{n}$ for every active coalition $C \in \pi^*$.

Let $\mathcal{C}$ be the set of active coalitions in π^*. We partition $\mathcal{C}$ into s groups by setting $\mathcal{C}^1 = \{C \in \mathcal{C} \mid C \cap B_1 \neq \emptyset\}$ and $\mathcal{C}^i = \{C \in \mathcal{C} \mid C \cap B_i \neq \emptyset, C \notin \mathcal{C}^j \text{ for } j < i\}$ for $i = 2, \ldots, s$. Note that every active coalition $C \in \pi^*$ intersects some coalition in π: otherwise we could add C to π. Therefore, each active coalition in π^* belongs to one of the sets $\mathcal{C}^1, \ldots, \mathcal{C}^s$. Also, by construction, the sets $\mathcal{C}^1, \ldots, \mathcal{C}^s$ are pairwise disjoint. Further, since the coalitions in $\mathcal{C}^i$ are pairwise disjoint and each of them intersects B_i, we have $|\mathcal{C}^i| \leq |B_i|$ for each $i = 1, \ldots, s$. Thus, we obtain

$$\#(\pi^*) = \sum_{i=1,\ldots,s} \sum_{C \in \mathcal{C}^i} |C| \leq \sum_{i=1,\ldots,s} \sum_{C \in \mathcal{C}^i} \sqrt{n} \leq \sum_{i=1,\ldots,s} |\mathcal{C}^i|\sqrt{n} \leq \sum_{i=1,\ldots,s} |B_i|\sqrt{n} \leq \#(\pi)\sqrt{n},$$

which is what we wanted to prove. ■

4.3 Nash Stability

We have shown that a-GASP does not not always admit a Nash stable assignment (Proposition 1). In fact, it is difficult to determine whether a Nash stable assignment exists; we omit the proof due to space constraints.

Theorem 8. *It is* NP*-complete to decide whether* a-GASP *admits a Nash stable assignment.*

However, we will now argue that if agents' preferences satisfy INC, DEC, or MIX, a Nash stable assignment always exists and can be computed efficiently.

Theorem 9. *If (N, A, P) is an instance of* a-GASP *that is increasing, decreasing, or mixed increasing-decreasing, a Nash stable assignment always exists and can be found in polynomial time.*

Proof. For increasing preferences, we can start by choosing an arbitrary individually rational assignment π (e.g., $\pi(i) = a_\emptyset$ for all $i \in N$). If π is not Nash stable, there exists an agent $i \in N$ with $\pi(i) = a_\emptyset$ and an activity $a_j \in A^*$ such that $(a, |\pi^j| + 1) \in S_i$. We can then modify π by setting $\pi(i) = a_j$; clearly, this assignment remains individually rational. If the resulting assignment is still not Nash stable, we can repeat this step. Since at each step the number of agents assigned to the void activity goes down by 1, this process stops after at most n steps.

For decreasing preferences, we proceed as follows. We consider the activities one by one; at step j, we consider activity a_j. Let $N_j \subseteq N$ be the set of agents that remain unassigned at the beginning of step j. Let $N_{j,\ell} = |\{i \in N_j \mid (a_j, \ell) \in S_i\}|$, and set $k = \max\{\ell \mid N_{j,\ell} \geq \ell\}$. Thus, k is the size of the largest group of currently unassigned agents that can be assigned to a_j. By our choice of k, the set N_j contains at most k agents that are willing to share a_j with $k + 1$ or more other agents. We assign all these agents to a_j; if the resulting coalition contains $\ell < k$ agents, we assign $k - \ell$ additional agents that approve of (a_j, k) to a_j (the existence of these $k - \ell$ agents is guaranteed by our choice of k). This completes the description of the j-th step. Note that no agent that remains unassigned after this step want to be assigned to a_j: indeed, this activity is currently shared among k agents, so if he were to join, the size of the group that is assigned to a_j would increase to $k + 1$, and none of the unassigned agents is willing to share a_j with $k + 1$ other agents. If some agents remain unassigned after n steps, we assign them to the void activity. To see that this assignment is Nash stable, consider an agent i assigned to the void activity. For each activity a_j he did not want to join the coalition of agents assigned to a_j during step j. Since the set of agents assigned to a_j did not change after step j, this is still the case.

For mixed decreasing-increasing instances, we first remove all activities in A^+ and apply our second algorithm to the remaining instance; we then consider the unassigned agents and assign them to activities in A^+ using the first algorithm. ■

We will now consider the problem of finding a Nash stable assignment that maximizes the number of agents assigned to a non-void activity. This problem admits an efficient algorithm if A^* consists of a single simple activity.

Theorem 10. *There exist a polynomial-time algorithm that given an instance (N, A, P) of* a-GASP *with $A^* = \{a\}$ finds a Nash stable assignment maximizing the number of agents assigned to a non-void activity, or decides that no Nash stable assignment exists.*

Proof. For each $k = n, \ldots, 0$, our algorithm decides whether there exists a Nash stable assignment π with $\#(\pi) = k$, and outputs the largest value of k for which this is the case.

For each $i \in N$, let $S'_i = S_i^{\downarrow a}$. For $k = n$ a Nash stable assignment π with $\#(\pi) = n$ exists if and only if $n \in S'_i$ for each $i \in N$. Assigning every agent to $a_\emptyset$ is Nash stable if and only if $1 \notin S'_i$ for each $i \in N$. Now we assume $1 \leq k \leq n - 1$ and set $U_1 = \{i \in N \mid k \in S'_i, k + 1 \notin S'_i\}$, $U_2 = \{i \in N \mid k \notin S'_i, k + 1 \in S'_i\}$, and $U_3 = \{i \in N \mid k \in S'_i, k + 1 \in S'_i\}$.

If $|U_1| + |U_3| < k$ there does not exist an individually stable assignment π with $\#(\pi) = k$. If $U_2 \neq \emptyset$ no Nash stable assignment π with $\#(\pi) = k$ can exist, since every agent from U_2 would be unhappy. If $|U_3| > k$ no Nash stable assignment π with $\#(\pi) = k$ can exist, since at least one agent in U_3 would not participate and thus would be unhappy. Thus, we can assume that $|U_1| + |U_3| \geq k$, $|U_3| \leq k$, $U_2 = \emptyset$. In this case we can construct a Nash stable assignment π by assigning all agents from U_3 and $k - |U_3|$ agents from U_1 to a. Since we have $\pi(i) \neq a_\emptyset$ for all $i \in U_2 \cup U_3$, no agent is unhappy. ■

5 Conclusions and Future Work

We have defined a new model for the selection of a number of group activities, discussed its connections with hedonic games, defined several stability notions, and for two of them, we have obtained several complexity results. A number of our results are positive: finding desirable assignments proves to be tractable for several restrictions of the problem that are meaningful in practice. Interesting directions for future work include exploring the complexity of computing other solution concepts for a-GASP and extending our results to the more general setting of GASP.

Acknowledgments We would like to thank the anonymous COMSOC referees for their very useful feedback. This research was supported by National Research Foundation (Singapore) under Research Fellowship NRF2009-08, by the project ComSoc (ANR-09-BLAN-0305-01), and by the Austrian Science Fund (P23724-G11 and P23829-N13). This project was initiated during the Dagstuhl seminar 12101 "Computation and Incentives in Social Choice", and the authors are very grateful to Dagstuhl for providing a great research environment and inspiration for this work. Part of this work was done when the second author was visiting Université Paris-Dauphine.

References

[1] H. Aziz, F. Brandt, and P. Harrenstein. Pareto optimality in coalition formation. In *SAGT*, pages 93–104, 2011.

[2] C. Ballester. NP-competeness in hedonic games. *Games and Economic Behavior*, 49:1–30, 2004.

[3] S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18:135–153, 2001.

[4] A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38:201–230, 2002.

[5] S. Brams and R. Sanver. Voting systems that combine approval and preference. In *The Mathematics of Preference, Choice, and Order: Essays in Honor of Peter C. Fishburn*, pages 215—237. Springer, 2009.

[6] G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory*. Morgan and Claypool, 2011.

[7] J. Lenstra, D. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Math. Program.*, 46:259–271, 1990.

Andreas Darmann
Universität Graz, Austria
Email: andreas.darmann@uni-graz.at

Edith Elkind
Nanyang Technological University, Singapore
Email: eelkind@ntu.edu.sg

Sascha Kurz
Universität Bayreuth, Germany
Email: Sascha.Kurz@uni-bayreuth.de

Jérôme Lang
LAMSADE – Université Paris-Dauphine, France
Email: lang@lamsade.dauphine.fr

Joachim Schauer
Universität Graz, Austria
Email: joachim.schauer@uni-graz.at

Gerhard Woeginger
TU Eindhoven, The Netherlands
Email: gwoegi@win.tue.nl

Cost-sharing of continuous knapsacks

Andreas Darmann and Christian Klamler[1]

Abstract

This paper provides a first insight into cost sharing rules for the continuous knapsack problem. Assuming a set of divisible items with weights from which a knapsack with a certain weight constraint is to be filled, different such (classes of) rules are discussed. Those - based on individual approvals of the items - optimally fill the knapsack and share the cost of the knapsack among the individuals. Using various reasonable properties of continuous knapsack cost sharing rules, we provide three characterization results.

1 Introduction

Cost allocation in combinatorial optimization problems has been intensively discussed in recent years (see [14] for a summary). The major focus has been on the minimum cost spanning tree problem, the earliest and most widely investigated cost sharing problem in this area (e.g. [3], [4], [10]). There the interest lies mainly in the fair division of the cost of creating a network in which each agent is connected directly or indirectly to a source. A second emphasis has been on scheduling and queuing problems, i.e., on the problem of optimally processing jobs of different lengths or weights on a single server (e.g. [8], [12], [13]).

The above problem of finding minimum cost spanning trees has a major advantage among combinatorial optimization problems. Its optimal solution can be found in polynomial time. Only then, i.e., in the case of finding such an optimal solution "quickly", does it seem to make sense to talk about fairly sharing the costs, because otherwise any changes to the setting could make it impossible to find the new cost allocation in reasonable time. The focus could only be on fixed solutions.

Among the combinatorial optimization problems, the knapsack problem is concerned with efficiently filling a weight-restricted knapsack with items from a set of items with possibly different weights and profits. Efficiency in that respect means maximizing some profit function based on the items' profits. In case of indivisible items, this problem is typically NP-hard. One exception is the continuous knapsack problem in which the items are divisible and therefore the solution could contain a certain fraction of one item.

In usual cost sharing problems such as the bankruptcy problem ([1], [16]) or the minimum cost spanning tree problem, "objective" preferences such as costs or claims play a major role in determining a fair cost allocation. This will be different in our framework, where we focus on the approval or disapproval of certain items by individuals ([5]). The social welfare of a set of items is simply defined by the total number of approvals for the single items in the set ([6]). This could be seen as a first step towards using (binary) preference information in determining a fair cost allocation.

The setting used in this paper can be summarized as follows: we start with a certain knapsack (a capacity, time interval, etc.) and a set of items over which individuals have

[1] We are greatful to Ulrich Pferschy, Daniel Eckert and three anonymous referees for their comments on a previous version of this paper.

binary preferences. Each of the items has a (possibly different) weight. First, the goal is to fill the knapsack such that social welfare, (i.e., the sum of approvals) is maximized. Then the attempt is to fairly divide the cost of the knapsack (or maintaining the capacity, or using the time) among the individuals.

As an example consider a multi-national research project that has some pre-determined cost. Space and/or time constraints might limit the number of researchers (out of a pool of potential candidates) that can participate. In addition, the possible candidates might be forced to use the provided resource for their specific research for different amounts of time. The potential financing countries of the research project might approve and disapprove of different researchers. The question now is how to select the set of researchers and how to distribute the cost among the participating countries.[2]

In principle we are concerned with sharing the cost of a selected set of non-rival items that provides different utilities or payoffs to the individuals. Cost allocation aspects in such a binary knapsack problem have been considered before by Dror [9] and certain rules such as the Shapley value or the equal charge method have been suggested. In this paper we want to introduce and characterize (a family of) possibly interesting continuous knapsack cost sharing rules.

The following section establishes the formal framework, defines the continuous knapsack problem, and introduces reasonable properties of continuous knapsack cost sharing rules. Section 3 first introduces a whole family of such rules and then focuses on two rules of which characterization results are provided. Section 4 concludes the paper.

2 Preliminaries

Let $\mathcal{N} = \{1, \ldots, n\}$ denote a set of individuals, and $I = \{1, \ldots, m\}$ a set of items. With each item $j \in I$, we associate a positive weight $w_j \in \mathbb{R}_+$. The weights are summarized by the vector $\omega \in \mathbb{R}_+^m$, where the j-th entry ω_j corresponds to w_j.
Each individual $i \in \mathcal{N}$ partitions the set I into a set A_i of items she approves of and a set of items she disapproves of. For $i \in \mathcal{N}$, the vector representation $a_i \in \{0,1\}^m$ turns out to be useful, where the j-th entry $a_{i,j} = 1$ if individual i approves of item j, and $a_{i,j} = 0$ if i disapproves of j. These vectors are captured by means of an $n \times m$ matrix A, whose rows correspond to the vectors a_i; i.e., $A = (a_{i,j})_{i \in \mathcal{N}, j \in I}$.
$A \ominus a_i$ denotes the matrix resulting from A by deleting the row corresponding to a_i. Let B be a $k \times m$ matrix for some $k \in \mathbb{N}$. For some $b \in \{0,1\}^m$, $B \oplus b$ is the $(k+1) \times m$ matrix created by concatenating to B a $(k+1)$-st row β and setting $\beta = b$.
For $j \in I$, let $\mathcal{N}_j$ be the set of individuals of $\mathcal{N}$ who approve of j, i.e., $\mathcal{N}_j = \{i \in \mathcal{N} : j \in A_i\}$. The value p_j of item $j \in I$ is defined as the number of individuals that approve of j. Formally, $p_j := |\{i \in \mathcal{N} : j \in A_i\}| = |\mathcal{N}_j|$.

Given a capacity constraint (or weight bound) W, we can represent a knapsack cost sharing problem as the quadruple $(\mathcal{N}, A, \omega, W)$. A solution to this problem assigns to each individual a cost share. However, one of the major problems in this combinatorial optimization exercise is its computational complexity, i.e., finding an optimal knapsack is NP-hard. Hence, we need to restrict ourselves to a special setting of the knapsack problem. Therefore we assume the items to be divisible, i.e., a solution may contain fractions of (at most) one item. This is called the continuous knapsack problem introduced in the following subsection.

[2] A "fraction" of a researcher could be seen as a part-time worker.

2.1 The continuous knapsack

The following definition introduces a well-known optimization problem:

Definition 2.1 *(Continuous Knapsack Problem)*

Given a set $I = \{1, \ldots, m\}$ of items, and, for each $j \in I$, positive real numbers p_j and w_j, the continuous knapsack problem is the following problem:[3]

$$\begin{aligned} &\max \textstyle\sum_{j\in I} p_j x_j \\ s.t. \quad &\textstyle\sum_{j\in I} w_j x_j \le W \\ &x_j \in [0,1] \end{aligned}$$

It is known that the continuous knapsack problem can be solved in polynomial time (see [11]). In what follows, we assume that the items are sorted in a way such that

$$\frac{p_1}{w_1} > \frac{p_2}{w_2} > \ldots > \frac{p_m}{w_m} \tag{1}$$

Note that in practice, the strict inequalities in (1) are not a limitation, since these may always be reached by arbitrarily small "perturbations" of the weights or by modifying the accuracy of measurement. In theory (compare [11]), inequality (1) ensures that the unique solution the entity chooses is determined by

$$x_j := \begin{cases} 1 & \text{for } j = 1, \ldots, s-1 \\ \frac{1}{w_s}(W - \sum_{i=1}^{s-1} w_i) & \text{for } j = s \\ 0 & \text{for } j > s \end{cases} \tag{2}$$

where s is defined by

$$\sum_{j=1}^{s-1} w_j < W \quad \text{and} \quad \sum_{j=1}^{s} w_s \ge W$$

The corresponding objective function value z is given by $z = \sum_{j\in I} p_j x_j = \sum_{j=1}^{s-1} p_j + \frac{p_s}{w_s}(W - \sum_{i=1}^{s-1} w_i)$.

Item s is called *split item.*[4] For an optimal solution $X = (x_1, x_2, \ldots, x_m)$, we abbreviate $X_+ = \{j \in I : x_j > 0\} = \{1, \ldots, s\}$. In what follows, and in order to simplify notation, x_j is identified with its value in the optimal solution of the considered continuous knapsack problem.

2.2 Dividing a continuous knapsack

Let the quadruple $(\mathcal{N}, A, \omega, W)$ be given. From the previous section we know that a solution can be calculated in polynomial time. Now, the goal is to divide the cost of the optimally packed knapsack among the individuals in a fair manner. In that respect, we first have to determine the cost of the knapsack. In this paper, we assume that every unit of weight imposes a cost of one, and therefore the total cost of the knapsack is equal to the weight

[3] In our approach we will focus on maximizing a sort of utilitarian social welfare given by the sum of approvals. This might, however, not be the only way to implement a fair solution. More egalitarian approaches could also be considered at that stage.

[4] Note that possibly $x_s = 1$ holds in the optimal solution. That is, the split item s is not necessarily "split", i.e., $0 < x_s < 1$ need not hold.

constraint W. However, dividing then the weight w_j by W for each $j \in I$ and setting $W = 1$ does not change the structure of the problem (and, in particular, the optimal solutions of the corresponding continuous knapsack problems are identical). Thus, in the major part of the paper it is assumed that $W = 1$. In that case, the continuous knapsack cost sharing problem is denoted by the triple $(\mathcal{N}, A, \omega)$, and we refer to the corresponding continuous knapsack problem as the pair (A, ω).

In general, a continuous knapsack cost sharing rule is a function $\phi : (\mathcal{N}, A, \omega, W) \to \mathbb{R}^n_+$. The i-th entry ϕ_i of ϕ is interpreted as the share of the cost that individual i has to carry.

In the following we define some desirable properties for a continuous knapsack cost sharing rule, trying to capture certain aspects of fairness.

Properties of cost sharing rules.

The first requirement – frequently used in the literature in various contexts – is that the total cost of the knapsack should be allocated exactly.

Efficiency: A cost allocation rule ϕ is *efficient*, if $\sum_{i=1}^n \phi_i(\mathcal{N}, A, \omega, W) = W$.

For the sake of readability, the remaining properties (except additivity) are defined for the case $W = 1$. However, the definitions coincide with the ones for the general case.
The second property, widely used e.g. in scheduling problems ([13]), represents the idea that voters should not benefit from "splitting" into several voters with disjoint sets of approved items (or, the other way round, in case their approved items are disjoint, "merging" into a single voter). At the same time, the remaining voters should not be disadvantaged if certain voters "split up" (or "merge"). In principle this should prevent the creation of fake identities, i.e., the individual possibility to manipulate the fair division process.[5]

To illustrate the idea of splitting, let voter i approve of items $1, 2, 3$. Replacing voter i by voters i_j approving of item j only, $1 \le j \le 3$, should have the result that the sum of the cost shares of the three voters i_j has to be equal to the cost share of voter i in the original problem. In the following definition, given a set of individuals $\mathcal{N}'$, $A'_{i'}$ refers to the set of approved items of $i' \in \mathcal{N}'$ (and $a'_{i'}$ denotes the corresponding vector of approvals).

Split-proofness: Let $i \in \mathcal{N}$. Let $\mathcal{N}' = (\mathcal{N} \setminus \{i\}) \cup \{i_1, \ldots, i_r\}$, such that sets A'_{i_ℓ} form a partition of A_i, i.e., $\biguplus_{\ell=1}^r A'_{i_\ell} = A_i$. Let $A' = A \oplus (a'_{i_1} \oplus \ldots \oplus a'_{i_r}) \ominus a_i$.
A cost allocation rule ϕ is called *split-proof*, if

- $\phi_i(\mathcal{N}, A, \omega) = \sum_{j=1}^{|A_i|} \phi_{i_j}(\mathcal{N}', A', \omega)$ and
- $\phi_h(\mathcal{N}', A', \omega) \le \phi_h(\mathcal{N}, A, \omega)$ for all $h \in \mathcal{N} \setminus \{i\}$

Remark. Note that for a split-proof rule ϕ, the first of the above conditions implies that $\sum_{h \in \mathcal{N} \setminus \{i\}} \phi_h(\mathcal{N}, A, \omega) = \sum_{h \in \mathcal{N} \setminus \{i\}} \phi_h(\mathcal{N}', A', \omega)$. Thus, the mild second condition implies that $\phi_h(\mathcal{N}', A', \omega) = \phi_h(\mathcal{N}, A, \omega)$ holds for all $h \in \mathcal{N} \setminus \{i\}$. To see this, assume that the share of an individual h becomes strictly smaller in problem $(\mathcal{N}', A', \omega)$. Then, for at least one $h' \in \mathcal{N} \setminus \{j\}$ we must have $\phi_{h'}(\mathcal{N}', A', \omega) > \phi_{h'}(\mathcal{N}, A, \omega)$, in contradiction to the above definition.

Since each of the following two properties refers to an instance $(\mathcal{N}, A, \omega)$, for the sake of brevity we write ϕ_i instead of $\phi_i(\mathcal{N}, A, \omega)$ for $i \in \mathcal{N}$.

[5] It has to be added though, that the property is probably less compelling in this setting compared to scheduling problems, as fake identities are not allowed to overlap with their (sets of) approvals.

The first property reflects the compelling idea, well-known in the literature, that the cost allocation should not depend on the label of the individual.

Anonymity: Let $i, i' \in \mathcal{N}$. A cost allocation rule ϕ is called *anonymous*, if $(A_i = A_{i'} \Rightarrow \phi_i = \phi_{i'})$.

The second requirement is similar to the usual dummy-property. It states that an individual who only approves of items not in the optimal solution, should not be charged. A "totally unhappy" individual should not be forced to carry the knapsack or contribute to its costs.

Dummy: If $x_j = 0$ for all $j \in A_i$, then $\phi_i = 0$.

The following property applies non-manipulability arguments to situations in which pairs of individuals, that only approve of one single item, try to improve their situation by switching their approvals. It requires their cost shares to be exactly the same, i.e., providing absolutely no incentive to get involved into such switches.

Switch-proofness: Given $(\mathcal{N}, A, \omega)$, let $A_i = \{j\}$, $A_{i'} = \{j'\}$ with $x_j = x_{j'} = 1$. Let $(\mathcal{N}, \tilde{A}, \omega)$ with $\tilde{a}_h = a_h$ for all $h \in \mathcal{N} \setminus \{i, i'\}$ and $\tilde{a}_f = a_g$ for $f, g \in \{i, i'\}$, $f \neq g$. Then $\phi_k(\mathcal{N}, A, \omega) = \phi_k(\mathcal{N}, \tilde{A}, \omega)$ for all $k \in \mathcal{N}$.

A further reasonable property requires the division process to be independent of a possible sequential structure, i.e., if the knapsack is divided into two different and smaller knapsacks that together have exactly the same weight constraint as before, then applying the sharing rule to each of the smaller knapsacks separately should lead to the same total cost share as applying the rule to the original knapsack. This property will be called additivity and has been used, e.g., by [7] w.r.t. rights problems.

Additivity: Let $W^{(1)}, W^{(2)} \in \mathbb{R}_+$ with $W^{(1)} + W^{(2)} = 1$. Let $\phi^{(1)} = \phi(\mathcal{N}, A, \omega, W^{(1)})$, and let $X^{(1)}$ be the optimal solution of $(A, \omega, W^{(1)})$. Let $\tilde{A} = (\tilde{a}_{ij})_{i \in \mathcal{N}, j \in I}$ such that, for $i \in \mathcal{N}$, $\tilde{a}_{ij} = 0$ if $x_j^{(1)} = 1$ and $\tilde{a}_{ij} = a_{ij}$ otherwise.
In addition, let $\tilde{\omega} \in \mathbb{R}_+^m$ such that $\tilde{\omega}_j = (1 - x_j^{(1)})\omega_j$ for $j \in X_+^{(1)}$ with $0 < x_j^{(1)} < 1$, and $\tilde{\omega}_j = \omega_j$ otherwise. Let $\phi^{(2)} = \phi(\mathcal{N}, \tilde{A}, \tilde{\omega}, W^{(2)})$. Then, ϕ is *additive*, if $\phi = \phi^{(1)} + \phi^{(2)}$.

The final property is concerned with the changes in the cost shares given a minimal weight-change of a non-split item contained in the optimal solution of the continuous knapsack problem, keeping the remaining weights unchanged. It is exclusively concerned with situations in which everyone approves of exactly one item. A minimal weight change in that respect is one in which the optimal solution does not change, i.e., the set of items in the optimal solution before and after the weight change is identical.

Definition 2.2 *Given $(\mathcal{N}, A, \omega)$, let X be an optimal solution of the continuous knapsack problem (A, ω) with $X_+ = \{1, \ldots, s\}$ and $x_s < 1$. For some $j < s$, let $\tilde{w}_j < w_j$ and $\tilde{\omega} = (w_1, \ldots, w_{j-1}, \tilde{w}_j, w_{j+1}, \ldots, w_m)$.*
We call $\tilde{w}_j$ insignificantly smaller than w_j, if for the optimal solution $\tilde{X}$ of $(A, \tilde{\omega})$, we have $\tilde{X}_+ = X_+$.

Now, let the weight of j insignificantly decrease in the sense of the above definition, and let each individual approve of exactly one item. Then, weight-monotonicity states that all those that approve of the item that became insignificantly smaller should face a decrease in their cost share relative to the change in the value of the objective function. The formal definition of this condition is as follows:

Weight-monotonicity: Let $\tilde{w}_j$ be insignificantly smaller than w_j. Then, for all $i \in \mathcal{N}$ with $A_i = \{j\}$, $\frac{\phi_i(\mathcal{N}, A, \tilde{\omega})}{\phi_i(\mathcal{N}, A, \omega)} = \frac{z}{\tilde{z}}$, where $\tilde{z}$ denotes the objective function value of the optimal solution of $(A, \tilde{\omega})$.

3 Characterizations

In what follows, we consider a continuous knapsack cost sharing problem $(\mathcal{N}, A, \omega)$ where (as previously) X with $X_+ = \{1, \ldots, s\}$ corresponds to the optimal solution of the continuous knapsack problem (A, ω).

We now want to investigate, whether certain combinations of the previous properties can be used to determine specific reasonable cost sharing rules. Our first result establishes a full description of the family of efficient rules, that satisfies the dummy property, split-proofness and switch-proofness. As a second result, we present the characterization of a special representative of this family by adding weight-monotonicity. Finally, a characterization of another reasonable cost sharing rule is given.

Theorem 3.1 *The efficient rules that satisfy the dummy property, split-proofness and switch-proofness are exactly the functions ϕ^c with $0 \le c \le \frac{1}{\sum_{i<s} p_i}$, defined by ($\forall i \in \mathcal{N}$)*

$$\phi_i^c(\mathcal{N}, A, \omega) = c \cdot \sum_{j \in A_i} x_j + \mathbb{1}_{A_i}(s) \cdot \frac{1 - cz}{p_s}$$

Proof. First, we show that $\phi_i^c \ge 0$ holds for all $i \in \mathcal{N}$, i.e., ϕ^c is indeed a cost sharing rule. Since $c \ge 0$ holds, we obviously have $\phi_i^c \ge 0$ for i with $s \notin A_i$. If $s \in A_i$, then

$$\begin{aligned} \phi_i^c &= \textstyle\sum_{j \in A_i \setminus \{s\}} x_j c + x_s c + \frac{1-cz}{p_s} &= \textstyle\sum_{j \in A_i \setminus \{s\}} x_j c + x_s c + \left(\frac{1 - c\sum_{i=1}^{s-1} p_i - c p_s x_s}{p_s}\right) \\ &= \textstyle\sum_{j \in A_i \setminus \{s\}} x_j c + \left(\frac{1 - c\sum_{i=1}^{s-1} p_i}{p_s}\right) \end{aligned}$$

Due to $c \ge 0$, we have $\sum_{j \in A_i \setminus \{s\}} x_j c \ge 0$; in addition, $1 - c\sum_{i=1}^{s-1} p_i \ge 0$ holds because of $c \le \frac{1}{\sum_{i=1}^{s-1} p_i}$. Thus, $\phi_i^c \ge 0$ holds in the case $s \in A_i$ as well.

Now, it is shown that each of the axioms is satisfied by the proposed rule.
The dummy property is obviously satisfied. Now, consider $\sum_{i \in \mathcal{N}} \phi_i^c = \sum_{i \in \mathcal{N}} c \sum_{j \in A_i} x_j + \sum_{i \in \mathcal{N}} \mathbb{1}_{A_i}(s) \frac{1}{p_s}(1 - cz) = c \sum_{i \in \mathcal{N}} \sum_{j \in A_i} x_j + \frac{1}{p_s}(1 - cz) \sum_{i \in \mathcal{N}} \mathbb{1}_{A_i}(s)$. Since item j is approved by exactly p_j individuals of $\mathcal{N}$, it holds that $\sum_{i \in \mathcal{N}} \sum_{j \in A_i} x_j = \sum_{j \in I} p_j x_j = z$, and $\sum_{i \in \mathcal{N}} \mathbb{1}_{A_i}(s) = p_s$. Hence, $\sum_{i=1}^n \phi_i^c = cz + \frac{1}{p_s}(1 - cz)p_s = 1$, which proves efficiency.
For a fixed $i \in \mathcal{N}$, let $(\mathcal{N}', A', \omega)$ be as described in the definition of split-proofness. Note that the optimal solution X' of (A', ω) is also the optimal solution of (A, ω), and the respective objective function values z' and z coincide. Thus,

$$\sum_{\ell=1}^r \phi_{i_\ell}^c(\mathcal{N}', A', \omega) = \sum_{\ell=1}^r (c \sum_{j \in A'_{i_\ell}} x_j + \mathbb{1}_{A'_{i_\ell}}(s) \frac{1}{p_s}(1 - cz)) = c \sum_{\ell=1}^r \sum_{j \in A'_{i_\ell}} x_j + \frac{1}{p_s}(1 - cz) \sum_{\ell=1}^r \mathbb{1}_{A'_{i_\ell}}(s)$$

By construction, $\sum_{\ell=1}^r \sum_{j \in A'_{i_\ell}} x_j = \sum_{j \in A_i} x_j$, and $\sum_{\ell=1}^r \mathbb{1}_{A'_{i_\ell}}(s) = \mathbb{1}_{A_i}(s)$. Hence, $\sum_{\ell=1}^r \phi_{i_\ell}^c(\mathcal{N}', A', \omega) = c \sum_{j \in A_i} x_j + \mathbb{1}_{A_i}(s) \frac{1}{p_s}(1 - cz) = \phi_i^c(\mathcal{N}, A, \omega)$. I.e., ϕ^c is split-proof.
For switch-proofness, let $A_i = \{j\}$ and $A_{i'} = \{j'\}$ such that $x_j = x_{j'} = 1$. Let $\tilde{A}$ be built from A because i and i' "switch" their items (as in the definition of switch-proofness). Then, $\phi_k(\mathcal{N}, A, \omega) = c = \phi_k(\mathcal{N}, \tilde{A}, \omega)$ for $k \in \{i, i'\}$, since the optimal solutions of (A, ω) and $(\tilde{A}, \omega)$ coincide. The latter fact obviously implies $\phi_k(\mathcal{N}, A, \omega) = \phi_k(\mathcal{N}, \tilde{A}, \omega)$ for all $k \in \mathcal{N} \setminus \{i, i'\}$ as well.

On the other hand, assume there is a rule ψ that satisfies the stated conditions. Now in order to create the new instance $(\mathcal{N}', A', \omega)$ from $(\mathcal{N}, A, \omega)$, replace each voter i with the voters $i_1, \ldots, i_{|A_i|}$ such that $|A'_{i_\ell}| = 1$ for each $1 \le \ell \le |A_i|$ and $\bigcup_{\ell=1}^{|A_i|} A'_{i_\ell} = A_i$. Because of split-proofness, we know that

$$\sum_{\ell=1}^{|A_i|} \psi_{i_\ell}(\mathcal{N}', A', \omega) = \psi_i(\mathcal{N}, A, \omega) \tag{3}$$

holds for each $i \in \mathcal{N}$.
Obviously, the optimal solutions of (A, ω) and (A', ω) coincide; let X be such an optimal solution, with $X_+ = \{1, \ldots, s\}$. Note that the objective function value is given by

$$z = p_1 x_1 + \ldots p_s x_s = p_1 + \ldots p_{s-1} + p_s x_s$$

First, we show that ψ is anonymous. Let $i, j \in \mathcal{N}$ with $A_i = A_j$. Starting with instance $(\mathcal{N}', A', \omega)$, create instance $(\mathcal{N}', \tilde{A}', \omega)$ by applying a "switch" between the individuals i_k and j_k, $k \in \{1, \ldots, |A_i|\}$, i.e., $\tilde{A}'_g = A'_h$ holds for $g, h \in \{i_k, j_k\}$. Now, switch-proofness and the fact that ψ is a function imply $\psi_{i_k}(\mathcal{N}, A', \omega) = \psi_{i_k}(\mathcal{N}, \tilde{A}', \omega) = \psi_{j_k}(\mathcal{N}, A', \omega)$ for all $k \in \{1, \ldots, |A_i|\}$. Thus, $\psi_i(\mathcal{N}, A, \omega) = \sum_{\ell=1}^{|A_i|} \psi_{i_\ell}(\mathcal{N}', A', \omega) = \sum_{\ell=1}^{|A_i|} \psi_{j_\ell}(\mathcal{N}', A', \omega) = \psi_j(\mathcal{N}, A, \omega)$ is satisfied; i.e., ψ is anonymous.
Let $i, i' \in \mathcal{N}'$ with $A'_i = \{j\}$, $A'_{i'} = \{j'\}$ and $j, j' < s$. Then, perform a switch between i and i' and call the new instance $(\mathcal{N}', A^*, \omega)$. Because of split-proofness, we can assume that the last two rows of each A and A^* correspond to a'_i and $a'_{i'}$ (in the same order). Note that in A^*, the row a'_i displays ${A^*}_{i'}$ and the row $a'_{i'}$ displays ${A^*}_i$ respectively. Thus, since ψ is a function, we must have $\psi_i(\mathcal{N}', A', \omega) = \psi_{i'}(\mathcal{N}', A^*, \omega)$. However, switch proofness yields that $\psi_{i'}(\mathcal{N}', A', \omega) = \psi_{i'}(\mathcal{N}', A^*, \omega)$. Hence, we must have $\psi_i(\mathcal{N}', A', \omega) = \psi_{i'}(\mathcal{N}', A', \omega)$. Therefore, for some $c \ge 0$, $\psi_{g'}(\mathcal{N}', A', \omega) = c$ must hold for all $g' \in \mathcal{N}'$ with $A'_{g'} = \{h'\}$ and $x_{h'} = 1$.
Anonymity together with the dummy property implies that, for some $c_s, c \in \mathbb{R}_+ \cup \{0\}$,

$$\psi_{i'}(\mathcal{N}', A', \omega) = \begin{cases} c_s & \text{if } A_{i'} = \{s\} \\ c & \text{if } A_{i'} = \{j' : j' < s\} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Efficiency yields

$$1 = \sum_{i' \in \mathcal{N}'} \psi_{i'}(\mathcal{N}', A', \omega) = \sum_{i' \in \mathcal{N}'_s} \psi_{i'} + \sum_{j<s} \sum_{i' \in \mathcal{N}'_j} \psi_{i'} \tag{5}$$

Note that, by construction, for each $j \in I$, $|\mathcal{N}'_j| = p_j$. Equation (5) can hence be rewritten as

$$1 = p_s c_s + c \cdot (p_1 + p_2 + \ldots + p_{s-1}) \tag{6}$$

Recall that $z = p_1 + p_2 + \ldots + p_{s-1} + x_s p_s$, or, equivalently, $\sum_{i=1}^{s-1} p_i = z - x_s p_s$. Substituting the last equality in (6), we get

$$\begin{array}{rrcl} & 1 - p_s c_s & = & c(z - x_s p_s) \\ \Leftrightarrow & c_s & = & \frac{1-cz}{p_s} + x_s c \end{array} \tag{7}$$

With (3) and (4), we get $\psi_i(\mathcal{N}, A, \omega) = \sum_{\ell=1}^{|A_i|} \psi_{i_\ell}(\mathcal{N}', A', \omega) = \sum_{j \in A_i \setminus \{s\}} x_j c + c_s \cdot \mathbb{1}_{A_i}(s)$. With (7), this yields

$$\psi_i(\mathcal{N}, A, \omega) = \begin{cases} \sum_{j \in A_i} x_j c & \text{if } s \notin A_i \\ \sum_{j \in A_i} x_j c + \frac{1-cz}{p_s} & \text{if } s \in A_i \end{cases}$$

Analogously to the beginning of the proof, it follows that $0 \leq c \leq \frac{1}{\sum_{i<s} p_i}$ must hold for ψ to be a cost sharing rule. Therewith, $\psi = \phi^c$. □

A representative of the above family of rules is derived from the idea, that a voter's cost share should exclusively depend on the total number of the items in the optimal knapsack she approves of, relative to the total number of approvals for the entire knapsack (in each case taking fractional values into account[6]). In particular, if someone likes twice as many items (included as a whole) from the knapsack than another individual, then she should also be given a cost share twice as high. Obviously this cost sharing rule is not concerned with weights of items or number of approvals for one specific item. Formally, this rule can be defined as follows:

Definition 3.1 *Given a problem $(\mathcal{N}, A, \omega)$, the simple proportional continuous knapsack cost sharing rule is defined as $(\forall i \in N)$*

$$\phi_i^{sol}(\mathcal{N}, A, \omega) = \frac{\sum_{j \in A_i} x_j}{z}$$

The rule ϕ^{sol} can be characterized as follows.

Theorem 3.2 *$\phi_i^{sol}(\mathcal{N}, A, \omega)$ is the only efficient and split-proof rule that satisfies dummy, switch-proofness, and weight-monotonicity.*

Proof. ϕ^{sol} belongs to the family ϕ^c (setting $c = \frac{1}{z}$. Hence, due to Theorem 3.1, it is sufficient to show that ϕ^{sol} is the only among the rules ϕ^c that satisfies weight-monotonicity. It is easy to verify that ϕ^{sol} satisfies weight-monotonicity. To proof the other direction, we follow the argumentation of the above proof. Consider instance $(\mathcal{N}', A', \omega)$ (of the above proof) and assume $x_s < 1$. Decrease the weight of item j from w_j insignificantly to $\tilde{w}_j$ for some $j < s$ such that $x_s' = 1$ in the optimal solution $\tilde{X}$ (with objective function value $\tilde{z}$ of $(A', \tilde{\omega})$, where $(\mathcal{N}', A', \tilde{\omega})$ denotes this new instance). Call the new shares (according to (4)) c_s' and c'; note that due to $x_s' = 1$, with analogous arguments as in the proof of Theorem 3.1, from switch-proofness we get $c_s' = c'$.
From efficiency, we thus get $1 = p_s c_s' + c' \cdot (p_1 + p_2 + \ldots + p_{s-1}) = c'(p_1 + p_2 + \ldots + p_s) = c' \cdot \tilde{z}$. Therewith, $c' = \frac{1}{\tilde{z}}$. Weight-monotonicity, however, implies $\frac{\psi_i(\mathcal{N}', A', \omega)}{\psi_i(\mathcal{N}', A', \tilde{\omega})} = \frac{c'}{c} = \frac{z}{\tilde{z}}$ for $i \in \mathcal{N}'$ with $A_i = \{j\}$. Hence, $c = \frac{1}{z}$ follows. Thus, ψ corresponds to ϕ^c with $c = \frac{1}{z}$, i.e., $\psi = \phi^{sol}$. □

The above rule puts its focus purely on the proportion of individual approvals to total approvals. This might seem unreasonable or inefficient in certain situations for two reasons: First, where extensive weight differences between the single items can be observed, a rule being sensitive to weights and weight changes might be preferable. Second, the more individuals approve of a certain item in the knapsack, the lower should probably be their cost share, if one assumes a non-rival good whose cost it imposes on the knapsack does not depend on the number of approvals. Hence, if we replace switch-proofness and weight-monotonicity with additivity, we characterize a rule, that takes into account the "inefficiency" $\frac{w_j}{p_j}$ of item $j \in I$ directly. The cost sharing rule is defined as follows:

Definition 3.2 *Given a problem $(\mathcal{N}, A, \omega)$, the weight-and-approval-based proportional continuous knapsack cost sharing rule is defined as $(\forall i \in N)$*

$$\phi_i^e(\mathcal{N}, A, \omega) = \sum_{j \in A_i} \frac{w_j}{p_j} x_j$$

[6] I.e, if a fraction of an item is included in the knapsack, then only the respective fraction of the approval is taken into account.

The rule ϕ^e can be characterized as follows:

Theorem 3.3 *$\phi_i^e(\mathcal{N}, A, \omega)$ is the only efficient and split-proof rule that satisfies dummy, anonymity, as well as additivity.*

Proof. For readability, we write ϕ instead of ϕ^e within this proof. We first show that all these axioms are satisfied by ϕ.

$$\sum_{i\in\mathcal{N}} \phi_i = \sum_{i\in\mathcal{N}} \sum_{j\in A_i} \frac{w_j}{p_j} x_j = \sum_{j\in I} \sum_{i\in\mathcal{N}_j} \frac{w_j}{p_j} x_j = \sum_{j\in I} p_j \frac{w_j}{p_j} x_j = \sum_{j\in I} w_j x_j$$

However, the last sum in the above expression corresponds to 1 because X is an optimal solution of (A, ω); thus, ϕ is efficient.

Split-proofness, dummy and anonymity are obviously satisfied.

For additivity, let $W^{(1)}, W^{(2)} \in \mathbb{R}_+$ with $W^{(1)} + W^{(2)} = 1$. Note that $x_j = 0$ implies $x_j^{(1)} = 0$ and $x_j^{(2)} = 0$. Thus, it is sufficient to consider the items $\{1, \ldots, s\}$. By construction, $X_+^{(1)} = \{1, ..., \ell\}$ for some $\ell \le s$.

Case 1: $x_\ell^{(1)} = 1$. By construction, this means that there is no voter that approves of any of the items $\{1, ..., \ell\}$ in instance $(\mathcal{N}, \tilde{A}, \tilde{\omega}, W^{(2)})$. Thus, $x_j^{(2)} = 0$ for all $1 \le j \le \ell$. Vice versa, we have $x_j^{(1)} = 0$ and $x_j^{(2)} = x_j$ for all $j \in \{\ell+1, \ldots, s\}$. In addition, $\tilde{w}_j = w_j$ holds for $j \in \{\ell+1, \ldots, s\}$. Hence, $\phi_i^{(1)} + \phi_i^{(2)} = \sum_{j\in A_i} \frac{w_j}{p_j} x_j^{(1)} + \sum_{j\in A_i} \frac{\tilde{w}_j}{p_j} x_j^{(2)} = \sum_{j\in A_i} \frac{w_j}{p_j} x_j = \phi_i$.

Case 2: $0 < x_\ell^{(1)} < 1$. Then, in instance $(\mathcal{N}, \tilde{A}, \tilde{\omega}, W^{(2)})$, each of the items $\{1, \ldots, \ell-1\}$ has zero approvals. Thus the ranking analogous to (1) (restricted to the remaining items) is

$$\frac{p_\ell}{\tilde{w}_\ell} > \frac{p_{\ell+1}}{\tilde{w}_{\ell+1}} > \ldots > \frac{p_s}{\tilde{w}_s} > \ldots > \frac{p_m}{\tilde{w}_m}$$

because the number of approvals of these items remains unchanged, and only the weight of item ℓ has decreased (compared to the original instance).

Case 2a: $\ell \neq s$. By the choice of x_s and $\tilde{w}_\ell$, $W^{(2)} = \sum_{k=\ell}^{s-1} \tilde{w}_\ell + x_s w_s$ must hold. Thus, $X_+^{(2)} = \{\ell, \ell+1, \ldots, s\}$, and $x_\ell^{(2)} = \ldots = x_{s-1}^{(2)} = 1$ and $x_s^{(2)} = x_s$. As in the above case, by construction for all $\ell+1 \le j \le s$ we have $\tilde{w}_j = w_j$. Note that for $j \neq \ell$, $x_j^{(1)} + x_j^{(2)} = x_j$. Thus, if $\ell \notin A_i$, we get

$$\phi_i^{(1)} + \phi_i^{(2)} = \sum_{j\in A_i} \frac{w_j}{p_j} x_j^{(1)} + \sum_{j\in A_i} \frac{\tilde{w}_j}{p_j} x_j^{(2)} = \sum_{j\in A_i} \frac{w_j}{p_j} (x_j^{(1)} + x_j^{(2)}) = \sum_{j\in A_i} \frac{w_j}{p_j} x_j = \phi_i \qquad (8)$$

Let $\ell \in A_i$. By construction, $\tilde{w}_\ell = (1 - x_\ell^{(1)}) w_\ell$. With $x_\ell^{(2)} = 1$, analogously to equation (8) we get

$$\begin{aligned}
\phi_i^{(1)} + \phi_i^{(2)} &= \textstyle\sum_{j\in A_i\setminus\{\ell\}} \frac{w_j}{p_j}(x_j^{(1)} + x_j^{(2)}) + \frac{w_\ell}{p_\ell} x_\ell^{(1)} + \frac{\tilde{w}_\ell}{p_\ell} x_\ell^{(2)} \\
&= \textstyle\sum_{j\in A_i\setminus\{\ell\}} \frac{w_j}{p_j} x_j + \frac{w_\ell}{p_\ell} x_\ell^{(1)} + \frac{w_\ell}{p_\ell}(1 - x_\ell^{(1)}) \\
&= \textstyle\sum_{j\in A_i} \frac{w_j}{p_j} x_j \\
&= \phi_i
\end{aligned}$$

Case 2b: $\ell = s$. For $1 \le j \le s-1$, we thus have $x_j^{(1)} = x_j = 1$ and $x_j^{(2)} = 0$.

By construction, $\tilde{w}_s = (1 - x_s^{(1)}) w_s$ and $W^{(2)} = (x_s - x_s^{(1)}) w_s$. Hence, $x_s^{(2)} = \frac{1}{\tilde{w}_s} W^{(2)} = \frac{1}{\tilde{w}_s}(x_s - x_s^{(1)}) w_s = \frac{x_s - x_s^{(1)}}{1 - x_s^{(1)}}$. As a consequence,

$$\frac{w_s}{p_s} x_s^{(1)} + \frac{\tilde{w}_s}{p_s} x_s^{(2)} = \frac{w_s}{p_s} \left(x_s^{(1)} + (1 - x_s^{(1)}) \frac{x_s - x_s^{(1)}}{1 - x_s^{(1)}} \right) = \frac{w_s}{p_s} x_s$$

Therewith, $\phi_i^{(1)} + \phi_i^{(2)} = \sum_{j \in A_i} \frac{w_j}{p_j} x_j = \phi_i$ holds in this case as well. I.e., ϕ is additive.

Assume there is a rule ψ that satisfies efficiency, split-proofness, dummy, anonymity, as well as additivity. As in the above proofs, create a new problem $(\mathcal{N}', A', \omega)$ from $(\mathcal{N}, A, \omega)$ by replacing each voter i with the voters $i_1, \dots, i_{|A_i|}$ such that $|A'_{i_\ell}| = 1$ for each $1 \leq \ell \leq |A_i|$ and $\bigcup_{\ell=1}^{|A_i|} A'_{i_\ell} = A_i$. Since ψ is split-proof, we get

$$\sum_{\ell=1}^{|A_i|} \psi_{i_\ell}(\mathcal{N}', A', \omega) = \psi_i(\mathcal{N}, A, \omega) \quad \text{for all } i \in \mathcal{N} \tag{9}$$

Since ψ is efficient and split-proof,

$$1 = \sum_{i \in \mathcal{N}} \psi_i(\mathcal{N}, A, \omega) = \sum_{i \in \mathcal{N}} \sum_{\ell=1}^{|A_i|} \psi_{i_\ell}(\mathcal{N}', A', \omega) = \sum_{k \in I} \sum_{i \in \mathcal{N}_k} \psi_i(\mathcal{N}', A', \omega) \tag{10}$$

Because ψ is anonymous, it holds that for each $j \in I$, $\psi_i(\mathcal{N}', A', \omega) = \psi_{i'}(\mathcal{N}', A', \omega) =: \delta_j$ for $i, i' \in \mathcal{N}_j$. Due to the dummy property, we have

$$\delta_j = 0 \ \forall j > s \tag{11}$$

Thus, (10) is equivalent to

$$1 = \sum_{j=1}^{s} p_j \cdot \delta_j \tag{12}$$

In what follows, we make use of additivity. In the first step, let $W^{(1)} = w_1$ and $W^{(2)} = \sum_{j=2}^{s-1} w_j + w_s x_s$. Then, the optimal solution of $(A', \omega, W^{(1)})$ is given by packing item 1 in the knapsack, i.e., $x_1^{(1)} = 1$ and $x_j^{(1)} = 0$ for $j > 1$. Anonymity implies that, for $j \in I$, there are $\delta_j^{(1)}, \delta_j^{(2)} \in \mathbb{R}_+$ such that $\delta_j^{(1)} = \psi_i(\mathcal{N}', A', \omega, W^{(1)})$ and $\delta_j^{(2)} = \psi_i(\mathcal{N}', \tilde{A}', \tilde{\omega}, W^{(2)})$ for $i \in \mathcal{N}_j$. Clearly, $\delta_j^{(1)} = 0$ if $j \geq 2$ because of the dummy property. Hence, anonymity and efficiency imply $p_1 \delta_1^{(1)} = w_1$, and thus $\delta_1^{(1)} = \frac{w_1}{p_1}$. By construction, in instance $(\mathcal{N}, A', \tilde{\omega}, W^{(2)})$, there is no voter who approves of item 1. By the dummy property, this means $\delta_1^{(2)} = 0$. Because of additivity, we have

$$\delta_1 = \delta_1^{(1)} + \delta_1^{(2)} = \frac{w_1}{p_1} \tag{13}$$

In the second step, let $W^{(1)} = w_1 + w_2$ and $W^{(2)} = \sum_{j=3}^{s-1} w_j + w_s x_s$. The optimal solution of $(A', \omega, W^{(1)})$ is $x_1^{(1)} = x_2^{(1)} = 1$ and $x_j^{(1)} = 0$ for $j > 2$. The dummy property yields $\delta_j^{(1)} = 0$ for $j > 2$. This fact and efficiency imply

$$w_1 + w_2 = p_1 \delta_1^{(1)} + p_2 \delta_2^{(1)} \tag{14}$$

Note that there is no voter who approves of one of the items $\{1, 2\}$ in instance $(\mathcal{N}, A', \tilde{\omega}, W^{(2)})$. Thus, $\delta_j^{(2)} = 0$ for $j \in \{1, 2\}$; because of additivity, this means $\delta_j = \delta_j^{(1)}$ for $j \in \{1, 2\}$. In particular, with $\delta_1 = \frac{w_1}{p_1}$ (see (13)), this turns equation (14) into

$$\begin{aligned} w_1 + w_2 &= p_1 \tfrac{w_1}{p_1} + p_2 \delta_2 \\ \Leftrightarrow \qquad \delta_2 &= \tfrac{w_2}{p_2} \end{aligned}$$

Repeating this argumentation, after a total of $s-1$ steps we have $\delta_k = \frac{w_k}{p_k}$ for all $1 \leq k \leq s-1$. Considering the instance $(\mathcal{N}, A', \omega)$, from (12) we know that $1 = \sum_{k=1}^{s} p_k \delta_k$ holds (due to efficiency). Thus, we have $1 = \sum_{k=1}^{s-1} w_k + p_s \delta_s$. On the other hand, $1 = \sum_{k=1}^{s-1} w_k + w_s x_s$ holds because of the choice of x_s (see 2). Combining the two last equalities yields $p_s \delta_s = w_s x_s$, and thus $\delta_s = \frac{w_s x_s}{p_s}$. With (11), we have

$$\delta_j = \begin{cases} \frac{w_j}{p_j} & \text{for } j < s \\ \frac{w_s}{p_s} x_s & \text{for } j = s \\ 0 & \text{for } j > s \end{cases}$$

Hence, equation (9) and the definition of δ_j imply $\psi_i(\mathcal{N}, A, \omega) = \sum_{j \in A_i} \frac{w_j}{p_j} x_j$. I.e., ψ and ϕ^e coincide. □

4 Conclusion

In this paper we have investigated cost sharing w.r.t. the continuous knapsack problem. Instead of costs or claims, we used the number of approvals to determine the optimal solution. To share the costs of the knapsack, we first introduced a whole family of cost sharing rules, and then provided explicit characterizations of two particular rules. The first rule assumed each item in the knapsack to impose the same cost, and made the individuals pay purely relative to their number of approved items. An interesting question in that respect would be to analyse the incentives to state one's true preferences. The second rule, however, was aware of both, the weight of the items in the knapsack and the number of individuals that approve of each item. It seems absolutely reasonable that those individuals who almost exclusively approve of items in the knapsack and/or approve of heavier items in the knapsack should carry a larger share of the cost. Based on various reasonable properties for continuous knapsack cost sharing rules, we provided characterization results for the two solution methods. Of course, the rules discussed in this paper are perhaps of an obvious kind, not taking too much care of the step of finding the optimal solution. However, many extensions seem possible and of interest for future research. On the one hand, further different - and probably less obvious - sharing rules could be introduced and analysed. On the other hand, more preference information, such as complete individual rankings, and - in addition - different types of objective functions could be used in the process of finding the optimal solution.

References

[1] Bergantinos, G. and E. Sanchez (2003): The proportional rule for problems with constraints and claims. *Mathematical Social Sciences*, 43 (2), 225–249.

[2] Bergantinos, G. and J.J. Vidal-Puga (2007): A fair rule in minimum cost spanning tree problems. *Journal of Economic Theory*, 137 (1), 326–352.

[3] Bird, C.J. (1976): On cost allocation for a spanning tree: a game theoretic approach. *Networks*, 6, 335–350.

[4] Bogomolnaia, A. and H. Moulin (2010): Sharing a minimal cost spanning tree: beyond the folk solution. *Games and Economic Behavior*, 69, 238–248.

[5] Brams, S.J. and P.C. Fishburn (1983): *Approval Voting*. Cambridge, Ma: Birkhäuser Boston.

[6] Brams, S.J., Kilgour, D.M. and M.R. Sanver (2007): A minimax procedure for electing committees. *Public Choice*, 132, 401–420.

[7] Chun, Y. (1988): The proportional solution for rights problems. *Mathematical Social Sciences*, 15, 231–246.

[8] Chun, Y. (2006): No-envy in queueing problems. *Economic Theory*, 29, 151–162.

[9] Dror, M. (1990): Cost allocation: the traveling salesman, bin packing, and the knapsack. *Applied Mathematics and Computation*, 35(2): 191–207.

[10] Dutta, B. and A. Kar (2004): Cost monotonicity, consistency, and minimum cost spanning tree games. *Games and Economic Behavior*, 48, 223–248.

[11] Kellerer, H., Pferschy, U., and D. Pisinger (2004): *Knapsack Problems*. Springer, Berlin.

[12] Maniquet, F. (2003): A characterization of the Shapley value in queueing problems. *Journal of Economic Theory*, 109, 90–103.

[13] Moulin, H. (2008): Proportional scheduling, split-proofness, and merge-proofness. *Games and Economic Behavior*, 63, 567–587.

[14] Moulin, H. (2011): Cost sharing in networks: some open questions. *mimeo*, Rice University.

[15] Moulin, H. and F. Laigret (2011): Equal-need sharing of a network under connectivity constraints. *Games and Economic Behavior*, 72, 314–320.

[16] Thomson, W. (2003): Axiomatic and game-theoretic analysis of bankruptcy and taxation problems: a survey. *Mathematical Social Sciences*, 45, 249–297.

Andreas Darmann
Institute of Public Economics
University of Graz, Universitätsstrasse 15/E4
8010 Graz, Austria
Email: `andreas.darmann@uni-graz.at`

Christian Klamler
Institute of Public Economics
University of Graz, Universitätsstrasse 15/E4
8010 Graz, Austria
Email: `christian.klamler@uni-graz.at`

Voting with Partial Information: What Questions to Ask?

Ning Ding and Fangzhen Lin

Abstract

Voting is a way to aggregate individual voters' preferences. Traditionally a voter's preference is represented by a total order on the set of candidates. However, sometimes one may not have a complete information about a voter's preference, and in this case, can only represent a voter's preference by a partial order. Given this framework, there has been work on computing the possible and necessary winners of a (partial) vote. In this paper, we take a step further, look at sets of questions to ask in order to determine the outcome of such a vote. Specifically, we call a set of questions a deciding set for a candidate if the outcome of the vote for the candidate is determined no matter how the questions are answered by the voters, and a possible winning (losing) set if there is a way to answer these questions to make the candidate a winner (loser) of the vote. We discuss some interesting properties about these sets of queries and prove some complexity results about them under some well-known voting rules such as plurality and Borda.

1 Introduction

Voting is a general way to aggregate preferences when a group of people need to make a common decision but have disagreements on which decision to take. Voting is traditionally studied in game theory and social choice theory. Recently it has attracted much attention in AI for various reasons, see for example the survey [Chevaleyre *et al.*, 2007].

Traditionally, a voter's preference is assumed to be a complete linear order over possible candidates (outcomes,or alternatives). One can easily imagine situations where this assumption is too strong, either because the voter herself cannot rank all of the possibilities linearly or because as an observer, we do not have a complete knowledge about her preferences. In fact, one of the well-known formalisms for representing agents' preferences in AI, CP-nets [Boutilier *et al.*, 2004], assumes agents' preferences are partial-ordered. In the context of voting, there has been work in this direction as well. Given a partial ordering for each voter, Konczak and Lang [2005] considered the problem of deciding whether a candidate is a *necessary winner* and *possible winner*. A necessary winner is a candidate who is always a winner in every possible completion of the given partial preference profile, while a possible winner is one who is a winner in some of the completions. The complexities of these two problems under a variety of voting rules, especially the so-called positional scoring rules, have been extensively studied [Pini *et al.*, 2007; Xia and Conitzer, 2011; Betzler and Britta, 2010; Baumeister and Rothe, 2010]. More recently, Conitzer *et al.* [2011] considered a notion of manipulations in voting with partial information.

In this paper, we continue this line of work. Given a voting context consisting of a set of candidates, a set of voters, and for each voter, a partial order on the candidates, we consider in general how much additional information is still needed in order to make a particular candidate a winner or loser under a voting rule. If the candidate is already a necessary winner or a necessary loser, then no additional information is needed. Otherwise, one may want to know which voter is crucial in deciding the outcome, and for that voter what would be the important questions to ask. These are obviously important issues to consider when doing voter preference solicitation, and should have some interesting applications. For instance, in an election, a candidate's team may want to know that given what they already know about a group of people, whether more knowledge about their voting preferences would make any differences to the outcome of the election.

This "additional information or knowledge" can come in many forms. Here we take it to be a set of pair-wise comparison questions [Conitzer, 2009] of the following form: voter i, which candidate do you prefer, a or b? We then consider sets of these questions that can settle the outcome for a candidate. There are at least two possible approaches here. A cautious approach looks for a set of questions such that no matter how these questions are answered by the voters will determine whether the candidate will win or lose. We call such a set of queries a deciding set for the candidate. This amounts to saying that as far as the candidate is concerned, if a question is not in a deciding set, then this question is irrelevant and can be ignored. It is thus not surprising that there is a unique minimal deciding set regardless of which voting rule to use.

The cautious approach makes sense when we want additional information that can decide the outcome for the candidate in question. If we want additional information that would make the candidate a winner (or a loser), then another notion may be more appropriate. Consider the case when we want a candidate to be a winner. Here we may be interested in a set of questions for which there are answers that would lead to the candidate being a winner (or loser), hoping that when voters are asked about these questions they will either indeed answer them as expected or that we can somehow influence them to answer them that way. We call such a set of questions a possible winning (or losing) set for the candidate. As can be expected, minimal possible winning (or losing) sets may not be unique. In contrast to our static notion of query sets, [Conitzer and Sandholm, 2002] defined the dynamic notion of *elicitation tree* and studied some basic problems related to that concept.

The rest of the paper is organized as follows. We first review some basic notions of voting with complete and partial information, and the notions of possible and necessary winners [Konczak and Lang, 2005]. We then define our notions of deciding sets, possible winning sets, and possible losing sets. We then prove some interesting properties about deciding sets, and consider how to compute minimal deciding set under various votin g rules. We next do the same for possible winning sets, and then conclude the paper.

2 Preliminaries

We assume a finite set $N = \{1, ..., n\}$ for voters (players, or agents), and a finite set O for candidates (outcomes, or alternatives). A *preference ordering* p_i of a voter i is a total (linear) order on O, and a preference profile p is a tuple of preference orderings, one for each voter.

A voting rule (method) f is a function from preferences profiles to non-empty sets of outcomes. For a preference profile p, $f(p)$ is the set of winners. When a single winner is desired, a tie-breaking rule can be used to select the one from $f(p)$. Or f is required to be single-valued. In social choice theory terminology, when $f(p)$ can be a set of outcomes, it is called a social choice correspondence, and when $f(p)$ is always single-valued, it is called a social choice function.

Most of the popular voting rules can be defined using a score vector $(s_1, s_2, \cdots, s_m)$, where m is the number of candidates, and $\forall i < m$, $s_i \geq s_{i+1}$. Given such a score vector, for each voter i and preference ordering p_i, the kth ranked candidate according to p_i receives the score s_k from the voter. Given a preference profile p, a candidate's score is then the sum of of the scores that she receives from each voter, and the winners are those that have the highest score. Such voting rules are called scoring rules.

For instance, the *plurality* voting rule uses the score vector $(1, 0, \cdots, 0)$, the *veto* rule uses the score vector $(1, \cdots, 1, 0)$, and the *Borda* rule uses the score vector $(m, m-1, \cdots, 1)$.

As mentioned in the introduction, we consider the situation when the preference ordering of a voter may not be total, either because the onlooker who is studying the voting does not have a complete knowledge of the voter's preference or that the voter herself is not certain of her own preferences.

Formally a *partial preference ordering* p_i of voter i is a partial order on the set O of candidates:

for each $o \in O$, $(o, o) \in p_i$ (reflexivity), if both (o_1, o_2) and (o_2, o_1) are in p_i, then $o_1 = o_2$ (antisymmetry), and if (o_1, o_2) and (o_2, o_3) are in p_i, then $(o_1, o_3) \in p_i$ (transitivity). A *partial preference profile* is then a tuple of partial preference orderings, one for each voter.

Given a partial preference ordering p_i, an *extension* of p_i is a partial preference ordering p'_i such that $p_i \subseteq p'_i$. An extension of p_i that is a total order is called a *completion* of p_i. Similarly, an extension of a partial preference profile p is a partial preference profile p' such that for each i, p'_i is an extension of p_i, and a completion of a partial preference profile p is a preference profile that is an extension of p.

Under a voting rule f, a candidate o is said to be a *necessary winner* of a partial preference profile p, if for all completion p' of p, $o \in f(p')$. If there exists such a completion, then o is said to be a *possible winner* [Konczak and Lang, 2005]. Furthermore, if o is not a possible winner, then we call o a necessary loser; and if o is not a necessary winner, then we call o a possible loser.

3 Deciding sets, possible winning sets, and possible losing sets of queries

As mentioned in the introduction, our interest in this paper is on getting additional information to decide the outcome of a vote. This additional information will be in the form of comparison queries [Conitzer, 2009] to voters.

Definition 1 *A (comparison) query to voter i is one of the form i:$\{a, b\}$ that asks i to rank candidates a and b.*

When presented with the query i:$\{a, b\}$, the voter i has to answer either "a" (she prefers a over b) or "b" (she prefers b over a).

Definition 2 *An answer to a set Q of questions is a function σ from Q to O such that for any i:$\{a, b\} \in Q$, $\sigma(i$:$\{a, b\}) \in \{a, b\}$.*

Intuitively, if an answer σ maps i:$\{a, b\}$ to "a", then the preference (a, b) ($a \geq b$) is added to voter i's partial preference ordering, and this may entail some new preferences for i, and may even lead to a contradiction. In the following, we require an answer to be consistent with the preferences that the voters already have.

Definition 3 *Let p be a partial preference profile and Q a set of queries. An answer σ to Q is legal under p if for each voter i, the transitive closure of the following set*

$$p_i \cup \{(a, b) \mid i\text{:}\{a, b\} \in Q \wedge \sigma(i\text{:}\{a, b\}) = a\}$$

which we denote by $p_i(\sigma, Q)$, is a partial order on O, the set of candidates. Given a legal answer σ to Q under p, the resulting partial preference profile is then

$$p(\sigma, Q) = (p_1(\sigma, Q), ..., p_n(\sigma, Q)),$$

In the following, unless stated otherwise, we always assume that answers to sets of questions are legal under the given partial preference profile.

We can now define the sets of questions that we are interested in this paper. A deciding set of queries for a candidate o determines the outcome of the vote for o no matter how the queries in the set are answered.

Definition 4 *Let p be a partial preference profile, o a candidate, and f a voting rule. A set Q of queries is a deciding set for o (in p under f) if for every answer σ, o is either a necessary winner or a necessary loser in the new partial profile $\sigma(p, Q)$ under f. Q is a minimal deciding set for o if it is a deciding set and there is no other deciding set Q' such that $Q' \subset Q$.*

Consider the incomplete profile in Table 1. If we take plurality as the voting rule, the minimal deciding set for candidate a is $\{2{:}\{a,b\}, 3{:}\{b,c\}\}$. Firstly, it is a deciding set: if $\sigma(2{:}\{a,b\}) = a$ then a is necessary winner; otherwise if $\sigma(2{:}\{a,b\}) = b$ and $\sigma(3{:}\{b,c\}) = c$, then a is also a necessary winner; and otherwise if $\sigma(2{:}\{a,b\}) = b$ and $\sigma(3{:}\{b,c\}) = b$, then a is a necessary loser.

Next we prove that all its proper subsets are not deciding sets. To prove this we only need to look at its subsets with size one. For $\{2{:}\{a,b\}\}$, a counterexample is when $\sigma(2{:}\{a,b\}) = b$. Given this answer, a is both a possible winner and a possible loser in the new partial preference profile. Similarly for $\{3{:}\{b,c\}\}$, we get a counterexample when $\sigma(3{:}\{b,c\}) = b$.

Notice here that the comparison queries 2:$\{a,c\}$ and 3:$\{a,b\}$ are not in the minimal deciding set.

1	a	$>$	b	$>$	c
2	b	$>$	c		
3	c	$>$	a		

Table 1: Partial preference profile

Sometimes one may also be interested in knowing the ways to make a candidate a winner or a loser in a vote. In this case, one may want to find sets of queries that when answered properly will lead to the candidate being a winner (or loser).

Definition 5 *Let p be a partial preference profile, o a candidate, and f a voting rule. A set Q of queries is a possible winning (losing) set for o (in p under f) if there is an answer σ such that o is a necessary winner (loser) in the new partial profile $\sigma(p,Q)$ under f. Q is a minimal possible winning (losing) set for o if it is a possible winning (losing) set for o, and there is no other possible winning (losing) set Q' for a such that $Q' \subset Q$.*

For the example in Table 1, if we still use plurality as the voting rule, then $Q_1 = \{2{:}\{a,b\}\}$ is a possible winning set for a to win because if we set $\sigma_1(2{:}\{a,b\}) = a$ then in the new partial profile $p(\sigma_1, Q_1)$ as shown in Table 2, a is a necessary winner. Notice that it is not a deciding set. And this possible winning set is obviously minimal because its only proper subset $\emptyset$ is not a possible winning set for a.

The set $Q_2 = \{3{:}\{a,b\}\}$ is also a minimal possible winning set because when $\sigma_2(3{:}\{a,b\}) = a$ as shown in Table 3, then in the new partial profile a is again a necessary winner. From this we can see that there could be multiple minimal possible winning sets for a candidate. Also notice that the query 3:$\{a,b\}$ is not in the minimal deciding set. So a minimal possible winning set may not have any overlap with the minimal deciding set.

1	a	$>$	b	$>$	c
2	a	$>$	b	$>$	c
3	c	$>$	a		

Table 2: $\sigma_1(p, Q_1)$

1	a	$>$	b	$>$	c
2	b	$>$	c		
3	c	$>$	a	$>$	b

Table 3: $\sigma_2(p, Q_2)$

It is easy to see that deciding sets always exist, and if Q is a deciding set, and $Q \subseteq Q'$, then Q' is also a deciding set. Furthermore, if $Q \neq \emptyset$, and Q is a deciding set for o, then Q is both a possible winning set and a possible losing set for a. But the converse is obviously not true in general.

In the following, we consider computing minimal deciding sets under the plurality and Borda rules. The case for the veto voting rule is similar to that of plurality.

4 Computing minimal deciding sets

If Q is a deciding set for candidate o, then for any query q not in Q, as far as the outcome for o is concerned, the answer to q is immaterial, thus can be totally ignored. This suggests that for any voting rule, any partial preference profile, and any candidate, there is a unique minimal deciding set for the candidate. This is indeed the case.

Theorem 1 *For any voting rule f, partial preference profile p, and candidate o, there is a unique minimal deciding set for o in p under f.*

To prove this theorem, we need the following lemma about partial orders.

Lemma 1 *Let R be a partial order on S, and $a \neq b$ two elements in S that are not comparable in R. Then there are two total orders R_1 and R_2 such that they both extend R, and are exactly the same except on a and b: for any x and y, $(x, y) \in R_1$ iff $(x, y) \in R_2$ provided $\{x, y\} \neq \{a, b\}$, and $(a, b) \in R_1$ but $(b, a) \in R_2$.*

Proof of Theorem Since the number of voters is finite, there exists a minimal deciding set Q for o. Let Q' be any other deciding set for o. If Q is not a subset of Q', then there is a $q \in Q$ but $q \notin Q'$. Let $Q_0 = Q \setminus \{q\}$. We show that Q_0 is also a deciding set. To show this, suppose σ is an answer to Q_0 under p. We need to show that o is either a necessary winner or a necessary loser in the new partial profile $p(\sigma, Q_0)$. Suppose q is $i{:}\{x, y\}$ for some voter i and candidates $x \neq y$. There are two cases:

1. The answer σ already entails an answer to q, that is, either (x, y) or (y, x) is in $p_i(\sigma, Q_0)$. This basically means that σ is also an answer to Q. Thus o must be either a necessary winner or a necessary loser in the new partial profile $p(\sigma, Q_0)$ as $p(\sigma, Q_0) = p(\sigma, Q)$ and Q is a deciding set.

2. Otherwise, by applying Lemma 1 to the partial order $p_i(\sigma, Q_0)$, we see that there are two answers σ_1 and σ_2 to $Q \cup Q'$ such that σ_1 and σ_2 are the same except on q where we have $\sigma_1(q) = x$ and $\sigma_2(q) = y$. Since $q \notin Q'$, σ_1 and σ_2 are the same answer when restricted to Q'. Since Q' is a deciding set, this means that o must be either a necessary winner in $p(\sigma_1, Q')$ or a necessary loser in $p(\sigma_1, Q')$. Suppose o is a necessary winner in $p(\sigma_1, Q')$. Then o is also a necessary winner in $p(\sigma_2, Q')$ as $p(\sigma_1, Q')$ is the same as $p(\sigma_2, Q')$. It follows then that o must also be a necessary winner in both $p(\sigma_1, Q \cup Q')$ and $p(\sigma_2, Q \cup Q')$. Since Q is also a deciding set, o is also a necessary winner in both $p(\sigma_1, Q)$ and $p(\sigma_2, Q)$. This means that o is a necessary winner in $p(\sigma, Q_0)$. Similarly, if a is a necessary loser in $p(\sigma_1, Q')$, then o is also a necessary loser in $p(\sigma, Q_0)$.

From this theorem, we get the following corollary.

Corollary 2 *If Q_1 and Q_2 are both deciding sets for a candidate o, then $Q_1 \cap Q_2$ is also a deciding set for o.*

Our next result provides a way to check if a query is in a minimal deciding set.

Suppose S is the set of all comparison queries. Then trivially, S is a deciding set for any candidate in any partial preference profile under any voting rule. Now consider any query $q \in S$, and any given candidate o and partial profile p. Since there is a unique minimal deciding set for o in p, it is clear that q is in the minimal deciding set iff $S \setminus \{q\}$ is not a deciding set.

We thus have the following proposition.

Proposition 1 *Let S be the set of all (comparison) queries. For any candidate o, and any partial preference profile p, a query $q = i\!:\!\{a, b\}$ is in the minimal deciding set if and only if there is an answer σ to $S \setminus \{q\}$ such that it can be extended to two answers σ_1 and σ_2 to S such that $\sigma_1(q) = a$, $\sigma_2(q) = b$, and the outcome of o is different in $p(\sigma_1, S)$ and $p(\sigma_2, S)$ (answer to the question "is o is necessary winner or loser?" is different).*

This proposition will be used in our algorithm for computing the minimal deciding sets under the plurality rule.

4.1 Plurality

For the plurality and the veto rules, computing the minimal deciding set can be done in polynomial time. We show this for the plurality rule.

Based on Proposition 1, it suffices to check each query independently. Now, for a given query $q = i{:}\{o_1, o_2\}$, one may think that we need to check if there are two extensions σ_1 and σ_2 that are different only for q, with a and b ranked top by i respectively, for every pair of candidates a and b. Actually as plurality only concerns the number of times a candidate is ranked first, the answer to q can affect the score vector only when o_1 and o_2 are ranked top in i's vote in σ_1 and σ_2. Now we come down to a problem of whether there exists an extension of all votes except i's such that when i's vote is considered, the "outcome" for the targeted candidate o changes (when i's top choice is o_1 or o_2). This problem is not solely a flow problem because it concerns the score of two candidates. However, it can be reduced to a flow problem, as shown in EqualScore procedure below. Here is a detailed description of our algorithm.

Our algorithm makes use of an algorithm for MAX-FLOW problem introduced in [Cormen *et al.*, 2001]. The problems is, given a graph with capacity as numbers assigned to every edge, to determine the maximal amount of flow going from node s to t with the flow in each edge not exceeding its capacity. Here we use MAX-FLOW(G, s, t) to denote the maximal flow from s to t in the flow graph G. Note that there are polynomial algorithms for MAX-FLOW(G, s, t).

Given a partial preference profile p, we use $a >_i b$ to stand for $(a, b) \in p_i$. When we add some new preferences $a >_i b, c >_i d$, etc, to p, we mean that we get a new partial preference profile p' such that $p'_j = p_j$ for every $j \neq i$, and p'_i is the transitive closure of $p_i \cup \{(a, b), (c, d), \cdots\}$. When we delete some voters $i_1, i_2, \cdots, i_k$ from p, we mean that we get a new partial preference profile p' such that the set of voters is $V \backslash \{i_1, i_2, \cdots, i_k\}$, and $p'_j = p_j$ for all $j \notin \{i_1, i_2, \cdots, i_k\}$. When we delete some candidates $o_1, o_2, \cdots, o_k$ from p, we mean that we get a new partial profile p' such that the set of candidates is $O' = O \backslash \{o_1, o_2, \cdots, o_k\}$, and p'_j is just p_j constrained to O'.

Algorithm:QueryInMDS($i{:}\{o_1, o_2\}, a, p$)

Input: a query $i{:}\{o_1, o_2\}$, a candidate a and a partial preference profile p.
Output: *yes* or *no* of whether $i{:}\{o_1, o_2\}$ is in the minimal deciding set of a in p.

1. If in p there is a w in $O \backslash \{o_1, o_2\}$ s.t. $w >_i o_1$ or $w >_i o_2$, then return *no*.
2. Else if $a \notin \{o_1, o_2\}$, then we do the following. First, let p^1 be the profile we get by adding $o_1 >_i o_2$ and $o_2 >_i c$ for every $c \notin \{o_1, o_2\}$ to p. If EqualScore(a, o_2, p^1) = *yes*, then return *yes*, else let p^2 be the profile we get by adding $o_2 >_i o_1$ and $o_1 >_i c$ for every $c \neq o_1, o_2$ into p. If EqualScore(a, o_1, p^2) = *yes*, then return *yes*, else return *no*.

3. Else, $a \in \{o_1, o_2\}$. W.L.O.G, let $a = o_1$. The case for $a = o_2$ is exactly the same. Let p^3 be the profile we get by adding $a >_i c$ for all alternative $c \neq a$ to p. If EqualScore$(a, m, p^3) =$ *yes* for some candidate $m \in O, m \neq a$, then return *yes*, else let p^4 be the profile we get by deleting voter i from p. p^4 has one less voter than p. If EqualScore$(a, o_2, p^4) =$ *yes*, then return *yes*, else return *no*.

Algorithm:EqualScore(a, b, p)

Input: candidates a and b and a partial preference profile p.
Output: *yes* or *no* of whether there is a completion p' of p s.t. the scores of a and b in p' are the same and the maximal among all candidates.

Let $S_a = \{i \mid \neg\exists w \in O\backslash\{a\}, w >_i a \in p\}$, $S_b = \{i \mid \neg\exists w \in O\backslash\{b\}, w >_i b \in p\}$, $S_i = S_a \cap S_b$, $s_a = |S_a|$, $s_b = |S_b|$, $s_i = |S_i|$.

1. If $|s_a - s_b| \leq s_i$ and $|s_a + s_b - s_i| \bmod 2 = 0$, then let p' be the new profile we get by deleting all the voters in $S_a \cup S_b$ and candidates a and b from p, and $T = |s_a + s_b - s_i|/2$. If Graph$(p', T) =$ *yes*, return *yes*, else return *no*.

2. Else if $|s_a - s_b| \leq s_i$ and $|s_a + s_b - s_i| \bmod 2 \neq 0$, then let $T = (|s_a + s_b - s_i| - 1)/2$. For every $i \in S_a \cup S_b$, let p' be the profile we get from deleting all the voters in $S_a \cup S_b \backslash \{i\}$ and candidates a and b from p. If Graph$(p', T) =$ *yes* then return *yes*. If none of these return *yes*, then return *no*.

3. Else we have $|s_a - s_b| > s_i$. W.L.O.G., let $s_b > s_a$. The case for $s_a > s_b$ is exactly the same. Let p' be the profile obtained by deleting all the votes in S_a and candidate a from p and set $T = s_a$. If Graph$(p', T) =$ *yes*, then return *yes*, else return *no*.

Algorithm:Graph(p, T).

Input: a partial preference profile p and a threshold T.
Output: *yes* or *no* of to indicate whether there is a completion p' of p, in which the maximal score of all candidates in p' is $\leq T$.

Let N be the set of voters and O the set of candidates in p. Let s and t be two new atoms not in $N \cup O$. Construct a flow graph G with $\{s, t\} \cup N \cup O$ as the set of nodes, and the following three layers of edges:

1. For every node in N, an edge from s to it with capacity one.

2. For every node $i \in N$ and every node $o \in O$ s.t. $\neg\exists o' \in O, o' >_i o$, an edge from i to o with capacity one.

3. For every $o \in O$, an edge from it to t with capacity T.

If MAX-FLOW$(G, s, t) = |N|$, then return *yes*, else return *no*.

Lemma 2 *Graph(p, T) returns yes iff p has a completion with every candidate getting at most score T under plurality.*

Lemma 3 *EqualScore(a, b, p) returns yes iff there is a completion p_c of p s.t. the scores of a and b in p_c are both the maximal score under plurality.*

We omit the proofs of these two lemmas here because of the page limit.

Corollary 3 *QueryInMDS$(i{:}\{o_1, o_2\}, a, p)$ returns yes iff $i{:}\{o_1, o_2\}$ is in the minimal deciding set of a in p under plurality, and it runs in polynomial time.*

The number of edges in the graph in procedure Graph is $O(mn)$, and the max flow found by Graph is $O(n)$. So if we use FORD-FULKERSON algorithm in [Cormen *et al.*, 2001] to implement MAX-FLOW, Graph runs in $O(mn^2)$ time. EqualScore calls Graph for at most n times, so the complexity of EqualScore is $O(mn^3)$. QueryInMDS calls EqualScore for $O(m)$ times, so QueryInMDS runs in $O(m^2n^3)$ time.

As plurality only concerns the candidate ranked first by every voter, according to Proposition 1, i:$\{o_1, o_2\}$ is in the minimal deciding set of a iff there are two assignments of a top choice for every voter, τ_1 and τ_2, which are consistent with p s.t. $\{\tau_1(i), \tau_2(i)\} = \{o_1, o_2\}$, and $\forall j \neq i, \tau_1(j) = \tau_2(j)$ and a is winner under assignment τ_1 but loser under τ_2. Firstly, we prove the "$\Rightarrow$" part of the corollary. In step 1, if the procedure does not return no then o_1 and o_2 are both legal top choices for i in p. In step 2, if the algorithm returns yes, then w.l.o.g we have EqualScore(a, o_2, p^1) = yes so there is an evidence τ_1 assigning the maximal number of votes to both a and o_2 with o_1 the top choice of voter i. This is just the evidence τ_1 in which a is a winner. And we can change $\tau_1(i)$ into o_2 to get τ_2, in which a is a loser. Notice that only τ_2 and τ_1 are different only on i. So $i:\{o_1, o_2\}$ is in the minimal deciding set of a. In step 3, the algorithm returns yes in two cases: EqualScore(a, m, p^3) is true for some $m \in O$ or EqualScore(a, o_2, p^4) is true. Here w.l.o.g we suppose $a = o_1$. If EqualScore(a, m, p^3) is true, then we have an assignment τ_1 in which a and m both have the maximal score among all candidates and $\tau_1(i) = a$. We can just change voter i's top choice from a into o_2 to get τ_2. And a is winner under τ_1 but loser under τ_2. So, again $i:\{o_1, o_2\}$ is in the minimal deciding set of a in p. If EqualScore(a, o_2, p^4) returns yes, then we have an assignment of candidates to every voter except i, such that a and o_2 have equal maximal score. This is an incomplete assignment of top choices τ'. Combining τ' with $\tau_1(i) = a$ we get τ_1 in which a wins, while combining it with $\tau_2(i) = o_2$ we get τ_2 in which a loses. So we can conclude i:$\{o_1, o_2\}$ is in the minimal deciding set of a.

Then we prove the "$\Leftarrow$" part of the corollary. As we just argued in the previous paragraph, there are two assignments of top choice τ_1 and τ_2 for every voter as we described. Suppose we record score of a candidate c under τ_1 and τ_2 as s_c^1 and s_c^2, and the maximal score among all candidates under τ_1 and τ_2 as $s_{\max}^1$ and $s_{\max}^2$. Notice that we always have $|s_a^1 - s_a^2| \leq 1, |s_{\max}^1 - s_{\max}^2| \leq 1$ and $s_a^1 = s_{\max}^1$ and $s_a^2 < s_{\max}^2$. If $a \notin \{o_1, o_2\}$, then $s_a^1 = s_a^2$, so $s_{\max}^2 - s_{\max}^1 = 1$, and $s_{\max}^1 = s_a^1$. As the $s_{\max}^2 > s_{\max}^1$ and w.l.o.g only o_2 has a higher score in τ_2 than in τ_1, so we have that $s_{o_2}^1 = s_{\max}^1 = s_a^1$ and $s_{o_2}^2 = s_{\max}^2$. Notice that under τ_1 which is consistent with p, o_2 and a both have maximal score among all candidates, so EqualScore$(a, o_2, p^2) =$ yes and so our algorithm will return yes in step 2.

If $a \in \{o_1, o_2\}$, then by a similar analysis we could conclude that our algorithm will also return yes. So we have proven Corollary 3. As there are only polynomial such queries, computing the minimal deciding set is also in P.

If $a \in \{o_1, o_2\}$, then w.l.o.g, we suppose $a = o_1$. Similarly, $s_a^1 = s_a^2 + 1$ and $s_{o_2}^2 = s_{o_2}^1 + 1$. Also, $|s_{\max}^1 - s_{\max}^2| \leq 1$ and $s_a^1 = s_{\max}^1$ and $s_a^2 < s_{\max}^2$. If a has unique maximal score in τ_1, then we have $s_{o_2}^1 = s_a^1 - 1$ because only o_2 has a higher score under τ_2 than under τ_1. So our algorithm will return yes because EqualScore$(a, o_2, p^4) =$ yes. If some other candidate m also has maximal score under τ_1, then it will also return yes because EqualScore$(a, m, p^3) =$ yes for candidate m.

4.2 Borda and other scoring voting rules

On the other hand, under the Borda voting rule and other scoring rules, computing minimal deciding sets is NP-complete.

The Borda voting rule uses the score vector $W = \{m, m-1, \cdots, 1\}$, where m is the number of candidates. Given a partial preference profile p, and a candidate o, it is known that checking if o is a possible winner in p under Borda is an NP-complete problem [Xia and Conitzer, 2011]. It came as no surprise that checking whether a query q is in the minimal deciding set is also an NP-complete problem.

Theorem 4 *The problem of checking if a query q is in the minimal deciding set for a candidate o in a partial profile p under the Borda voting rule is NP-complete.*

Proof The problem is in NP follows from Proposition 1 as checking whether an answer σ to $S \setminus \{q\}$ is legal in p, whether it can be extended to two different answers to q such that the outcomes of o in the two extensions are different under the Borda rule can all be done in polynomial time, where S is the set of all queries.

The problem is NP-hard because o is a possible winner iff either o is a necessary winner or the minimal deciding set for o is not empty. Notice that checking if o is a necessary winner under Borda can be done in polynomial time [Konczak and Lang, 2005].

In fact, the proof of this theorem gives a more general result:

Theorem 5 *For any polynomial time voting rule under which the possible winner problem is NP-complete and the necessary winner problem is in P, the problem of checking if a query is in the minimal deciding set is NP-complete.*

It is known that except for plurality and veto rules, all scoring rules have the property in the above theorem [Xia and Conitzer, 2011; Betzler and Britta, 2010; Baumeister and Rothe, 2010]. We thus have the following corollary.

Corollary 6 *For any scoring voting rule that is different from the plurality and the veto rules, checking if a query is in the minimal deciding set is NP-complete.*

5 Computing possible winning sets

We have seen that there may be multiple minimal possible winning sets. This makes the problem of computing these sets harder.

Proposition 2 *A candidate o is a possible winner in a partial preference profile p iff the set of all queries is a possible winning set for a. A candidate o is a possible loser iff she is not a necessary winner iff the set of all queries is a possible losing set for a.*

Thus just like Corollary 6, we have the following result.

Theorem 7 *For any scoring voting rule that is different from the plurality and the veto rules, checking if a set of queries is a possible winning set is NP-complete.*

We do not at present know the complexity of computing a minimal possible winning set. Our guess is that it is Π_2^P-complete, same as the complexity of computing minimal models (circumscription) in propositional logic [Eiter and Gottlob, 1993].

Neither do we know the exact complexity of checking if a set of queries is a possible losing set for a candidate. While Proposition 2 implies that checking if the set of all queries is a possible losing set for a candidate is in P, the problem seems to be harder in general as it requires checking whether an answer has enough information to conclude that the candidate is a necessary loser, which is a coNP-complete problem for voting rules such as Borda.

We now show that for the plurality voting rule, deciding whether a query set is a minimal possible winning set is in P.

5.1 Plurality

Algorithm: PossibleWinningSet(Q, p, a):

Input: A query set Q, a partial profile p and a candidate a, and we assume that $\forall i$:$\{b, c\} \in Q, (b, c) \notin p_i, (c, b) \notin p_i$.
Output: *yes* or *no* of whether Q is a possible winning set of a in p.

1. For every voter i, let G_i be the undirected graph with O as nodes and $\{(b, c) \mid i$:$\{b, c\} \in Q\}$ as edges and S_i be the set of all strongly connected components of G_i. For a strongly connected component u, we use $V(u)$ to represent the set of vertices of u. For every voter i, for every strongly connected component $u \in S_i$ s.t. $a \in V(u)$ and $\neg\exists o \in O, o >_i a$, add $a >_i c$ to p for every $c \neq a$ in $V(u)$. Set $s_a =$ the minimal score of a in p.

2. $\forall i \in N, U_i = \{u \in S_i \mid \forall o \in V(u), \neg\exists w \in O, w >_i o\}, U = U_1 \cup \cdots \cup U_n$

3. Let O be the set of candidates in p and U as defined. Let s and t be two new atoms not in $U \cup O$. Construct a graph G with $\{s, t\} \cup U \cup O \backslash \{a\}$ as set of nodes, and the following three layers of edges:

   (a) for every node in U an edge from s to it with capacity one.

   (b) for every node $u \in U$ and every candidate o s.t. $o \in O \backslash \{a\}$ and $o \in V(u)$, an edge from u to o with capacity one.

   (c) for every $o \in O \backslash \{a\}$ an edge from it to t with capacity s_a.

   If MAX-FLOW$(G, s, t) = |U|$, then return *yes*, else return *no*.

Corollary 8 *PossibleWinningSet*(Q, p, a) *returns yes iff* Q *is a possible winning set of* a *in* p*, and it runs in polynomial time.*

As proven in [Konczak and Lang, 2005], a is a necessary winner in p *iff* the minimal score of a is higher than the maximal score of any other candidate $c \in O$.

Intuitively, our algorithm tries to maximize the *min score* of a and see whether the *max score* of other candidates can be less than the min score of a under some answer of Q. The detailed proof of the correctness of the algorithm is omitted due to page limit.

The graph constructed has $O(mn)$ edges, and the flow found by the algorithm is $O(mn)$. So if we use FORD-FULKERSON alogrithm to implement MAX-FLOW, the flow calculation takes $O(m^2n^2)$ time. And the strongly connected components part runs in $O(m^2n)$ time as SCC is $O(|V| + |E|)$ and the size of the graph is $O(m^2)$. So PossibleWinningSet runs in $O(m^2n^2)$ time.

To determine whether Q is a minimal possible winning set, we just need $|Q|+1$ calls of the above algorithm. So determining whether a query set is a minimal possible winning set under plurality is also in P.

6 Related works

We have mentioned that this work generalizes the notions of necessary and possible winners [Konczak and Lang, 2005]. It is also closely related to work on vote elicitation (e.g. [Conitzer and Sandholm, 2002; Procaccia, 2008]). In vote elicitation, one is often interested in a dynamic question and answering process [Conitzer and Sandholm, 2002]. Here we are looking at statically, in the current state, how many possible questions one needs to ask in order to determine the outcome of a vote w.r.t. to a particular candidate. These two approaches are closely related. For instance, given our notion of minimal deciding sets, we can proceed in the following way to decide the outcome for a candidate x: in the current state, find a query that is in the minimal deciding set of x, ask the query and add the answer to the current partial preference profile; repeat this in the new state until one reaches a state where x is either a necessary winner or a necessary loser. The dynamic process thus obtained seems to be new, and we plan to explore its properties and connections with existing dynamic approaches in our future work.

Following the theoretical study of vote elicitation, researchers are recently doing some experimental studies of elicitation processes (e.g. [Lu and Boutilier, 2011a; 2011b; Kalech *et al.*, 2011]). In the these work, the focus is mainly to save the number of questions in and rounds of the elicitation process, and to develop approximations when the partial information is not enough to decide the winner. In contrast to these approaches, ours may be more easily parallelised and more efficient when we only care about one candidate.

7 Concluding remarks

We have considered sets of questions to ask the voters about in order to determine the outcome of a vote with partial information.

A deciding set is one that will determine the outcome of a vote for a candidate no matter how the queries in the set are answered. One fundamental property about this notion is that among these sets, there is a unique minimal one. Thus as far as a candidate is concerned, a comparison between two candidates is irrelevant to her if the associated query is not in her minimal deciding set. Computationally we have shown that the minimal deciding set can be computed in polynomial time for the plurality and veto rules, and is NP-complete to compute for other scoring rules. We will study complexity for other voting rules in our future work.

On the other hand, a possible winning (losing) set for a candidate is one that has an answer that will lead to the candidate being a necessary winner (loser). For a manipulator, these sets may be of more interest as they could tell her how to influence the voters to make the candidate a winner or a loser of the vote. We have shown that for plurality and veto rules, a minimal possible winning set can be computed in polynomial time. We believe that the same is true for computing a minimal possible losing set as well. For scoring voting rules such as Borda, the problem is again NP-hard for checking if a set of queries is a possible winning set.

References

[Baumeister and Rothe, 2010] Dorothea Baumeister and Jörg Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. In *ECAI-10*, pages 1019–1020, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[Betzler and Britta, 2010] Nadja Betzler and Dorn Britta. Towards a dichotomy for the possible winner problem in elections based on scording rule. *Journal of Computer and System Sciences*, 76:812–836, 2010.

[Boutilier *et al.*, 2004] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Cp-nets: A tool for representing and reasoning with conditional *Ceteris Paribus* preference statements. *JAIR*, 21:135–191, 2004.

[Chevaleyre *et al.*, 2007] Yann Chevaleyre, Ulle Endriss, Jrme Lang, and Nicolas Maudet. A short introduction to computational social choice. In Jan van Leeuwen, Giuseppe Italiano, Wiebe van der Hoek, Christoph Meinel, Harald Sack, and Frantiek Plil, editors, *SOFSEM 2007: Theory and Practice of Computer Science*, volume 4362 of *Lecture Notes in Computer Science*, pages 51–69. Springer Berlin / Heidelberg, 2007.

[Conitzer and Sandholm, 2002] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *AAAI-02*, pages 392–397. AAAI, 2002.

[Conitzer *et al.*, 2011] Vincent Conitzer, Toby Walsh, and Lirong Xia. Dominating manipulations in voting with partial information. In *IJCAI-2011 Workshop on Social Choice and Artificial Intelligence*, 2011.

[Conitzer, 2009] Vincent Conitzer. Eliciting single-peaked preferences using comparison queries, 2009.

[Cormen *et al.*, 2001] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*, chapter 26. MIT Press, 2001.

[Eiter and Gottlob, 1993] Thomas Eiter and Georg Gottlob. Propositional circumscription and extended closed world reasoning are Π_2^p-complete. *Theoretical Computer Science*, 114:231–245, 1993.

[Kalech *et al.*, 2011] Meir Kalech, Sarit Kraus, Gal A. Kaminka, and Claudia V. Goldman. Practical voting rules with partial information. *Autonomous Agents and Multi-Agent Systems*, 22(1):151–182, January 2011.

[Konczak and Lang, 2005] Kathrin Konczak and Jerome Lang. Voting procedures with incomplete preferences. In *Proc. IJCAI- 05 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[Lu and Boutilier, 2011a] Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *IJCAI-11*, pages 287–293, 2011.

[Lu and Boutilier, 2011b] Tyler Lu and Craig Boutilier. Vote elicitation with probabilistic preference models: empirical estimation and cost tradeoffs. In *Proceedings of the Second international conference on Algorithmic decision theory*, ADT'11, pages 135–149, Berlin, Heidelberg, 2011. Springer-Verlag.

[Pini *et al.*, 2007] M.S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. IJCAI-07, pages 1464–1469, 2007.

[Procaccia, 2008] Ariel D. Procaccia. *Computational Voting Theory: Of the Agents, By the Agents, For the Agents.* PhD thesis, The Hebrew University of Jerusalem, Sep. 2008.

[Xia and Conitzer, 2011] Lirong Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *JAIR*, 41:25–67, May 2011.

Ning Ding
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
Email: `dning@cse.ust.hk`

Fangzhen Lin
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
Email: `flin@cse.ust.hk`

Being Caught Between a Rock and a Hard Place in an Election—Voter Deterrence by Deletion of Candidates

Britta Dorn and Dominikus Krüger

Abstract

We introduce a new problem modeling voter deterrence by deletion of candidates in elections: In an election, the removal of certain candidates might deter some of the voters from casting their votes, and the lower turnout then could cause a preferred candidate to win the election. This is a special case of the variant in the family of 'control' problems in which an external agent is allowed to delete candidates and votes in order to make his preferred candidate win, and a generalization of the variant where candidates are deleted, but no votes. We initiate a study of the computational complexity of this problem for several voting systems and obtain $\mathcal{NP}$-completeness and $\mathcal{W}[2]$-hardness with respect to the parameter *number of deleted candidates* for most of them.

1 Introduction

Imagine: finally, you have the chance of getting rid of your old mayor, whom you absolutely cannot stand. Luckily, in addition to the normal unscrupulous opponents, the perfect candidate is running for the vote this year. You agree with everything he says and therefore you are even looking forward to Election Day. But suddenly the word is spread that he has withdrawn his candidacy. Again, you are feeling caught between a rock and a hard place. Does it make any sense to go to the polls if you only have a choice between the lesser of two evils?

Low voter turnouts caused by scenarios such as the one in the above example may lead to modified outcomes of an election. This is reminiscent of a family of problems which has been studied extensively in the computational social choice literature recently, the family of 'control' problems [1, 10–12, 17] where an external agent can change the outcome of an election by adding or deleting candidates and/or voters, respectively. In particular, in the setting of constructive control by deleting candidates, the agent can prevent candidates from running for office, which causes other candidates to rise in ranking for certain voters. This may ultimately result in the external agent's preferred candidate winning the election.

In real life, this process is a little bit more complicated and control of an election can occur in a more entangled way: As in our introductory example, if some candidates do not stand for election, then certain voters will not even take part in the election because they feel that there is nothing interesting to decide or no relevant candidate to vote for. The lower turnout could have consequences for the remaining candidates: the winner of the election under normal conditions might lose points because of the lower polling after the deletion of certain candidates, and this can produce a different winner. Hence, by *deterring* the voters by means of deleting their favorite candidates, one might prevent them from casting their votes and therefore change the outcome of the election. Therefore, we call this phenomenon *voter deterrence*.

This situation can be observed in the primaries in US elections or in mayoral elections, where mayors often are elected with single-digit turnout, sometimes caused by the withdrawal of candidacy of one or several alternatives in the run-up.

As to our knowledge, this problem has not yet been considered from a computational point of view. In this paper, we want to initiate the study of the corresponding decision problem Voter Deterrence defined below. We mainly consider the case where voters are easily deterred: As soon as their most preferred candidate does not participate in the election, they refrain from the election. This is what we denote as 1-Voter Deterrence, but clearly, one can also consider x-Voter Deterrence, where a voter only refuses to cast his vote if his top x candidates are removed. Surprisingly, it turns out that 1-Voter Deterrence is already computationally hard for several voting systems, even for Veto.

This paper is organized as follows. After introducing notation and defining the decision problem x-Voter Deterrence in Section 2, we investigate the complexity of this problem for the case of $x = 1$ for the voting systems Plurality (for which it turns out to be solvable in polynomial time, but it is $\mathcal{NP}$-complete for $x = 2$), Veto, 2-approval, Borda, Maximin, Bucklin, Fallback Voting, and Copeland (for all of which the problem turns out to be $\mathcal{NP}$-complete). As a corollary, we can show that the hard problems are also $\mathcal{W}[2]$-hard with respect to the solution size, i.e., with respect to the parameter *number of deleted candidates*, which means that they remain hard even if only few candidates have to be deleted to make the preferred candidate win. This is stated in Section 4 together with a short discussion of the complexity with respect to the parameter *number of candidates*. We conclude with a discussion of open problems and further directions that might be interesting for future investigations.

2 Preliminaries

Elections. An *election* is a pair $E = (C, V)$ consisting of a *candidate* set $C = \{c_1, \ldots, c_m\}$ and a multiset $V = \{v_1, \ldots, v_n\}$ of *votes* or *voters*, each of them a linear order over C, i.e., a transitive, antisymmetric, and total relation over the candidates in C, which we denote by $\succ$. A *voting system* maps (C, V) to a set $W \subseteq C$ called the *winners* of the election. All our results are given for the *unique winner case*, where W consists of a single candidate.

We will consider the voting systems Plurality, Veto, 2-approval, Borda, Maximin, Bucklin, Fallback Voting, and Copeland. A description of these systems can be found e.g. in [6].

Voter Deterrence, Control. In an x-Voter Deterrence instance, we are given an election $E = (C, V)$, a preferred candidate $p \in C$, and natural numbers k, $x \leq |C|$, as well as a voting system. It will always be clear from the context which voting system we are using, so we will not mention it explicitly in the problem description. Let $R \subseteq C$ denote a subset of candidates, and let $V_R \subseteq V$ denote the set of voters who have ranked only candidates from R among the first x ranks in their vote. The task consists in determining a set R of at most k candidates that are removed from C, and who therefore prevent the set of voters V_R from casting their votes, such that p is a winner in the election $\widetilde{E} = (C \setminus R, V \setminus V_R)$. The set R is then called a *solution* to the x-Voter Deterrence instance. The underlying decision problem is the following.

x-Voter Deterrence
Given: An election $E = (C, V)$, a preferred candidate $p \in C$, and k, $x \in \mathbb{N}$.
Question: Is there a subset of candidates $R \subseteq C$ with $|R| \leq k$, such that p is the winner in the election $\widetilde{E} = (C \setminus R, V \setminus V_R)$?

x-Voter Deterrence is a special case of one of the many variants in the family of 'control' problems [11], where the chair is allowed to delete candidates and votes, which is defined as follows.

Constructive Control by Deleting Candidates and Votes
Given: An election $E = (C, V)$, a preferred candidate $p \in C$, and k, $l \in \mathbb{N}$.

Question: Is there a subset $C' \subseteq C$ with $|C'| \leq k$, and a subset $V' \subseteq V$ with $|V'| \leq l$, such that p is a winner in the election $\widetilde{E} = (C \setminus C', V \setminus V')$?

Note that in the VOTER DETERRENCE problem, the deleted candidates and votes are coupled, which is not necessarily the case in the above control problem. In [11], it is shown that the above control problem is $\mathcal{NP}$-hard for the voting systems Plurality, Condorcet, Copeland$^\alpha$ ($0 \leq \alpha \leq 1$), Approval voting, and Maximin. However, since x-VOTER DETERRENCE is a special case of this variant of control, this does not settle its complexity for these voting systems.

If we set $x = m$, we obtain CONSTRUCTIVE CONTROL VIA DELETING CANDIDATES, which is the above control problem with $l = 0$. The latter variant hence is a special case of m-VOTER DETERRENCE, implying that the hardness results from [1, 12] carry over, i.e., m-VOTER DETERRENCE is $\mathcal{NP}$-hard for Plurality and Copeland$^\alpha$ for $0 \leq \alpha \leq 1$.

In this paper, we will mainly consider 1-VOTER DETERRENCE, i.e., a voter will refuse to cast his vote if his most preferred candidate does not participate in the election. For the voting system Plurality, we also consider 2-VOTER DETERRENCE, where a voter only refrains from voting if his two top ranked candidates are eliminated from the election.

Parameterized complexity. The computational complexity of a problem is usually studied with respect to the size of the input I of the problem. One can also consider the parameterized complexity [8, 15, 18] taking additionally into account the size of a so-called parameter k which is a certain part of the input, such as the number of candidates, or the size of the solution set. A problem is called *fixed-parameter tractable* with respect to a parameter k if it can be solved in $f(k) \cdot |I|^{O(1)}$ time, where f is an arbitrary computable function depending on k only. The corresponding complexity class consisting of all problems that are fixed-parameter tractable with respect to a certain parameter is called $\mathcal{FPT}$.

The first two levels of (presumable) parameterized intractability are captured by the complexity classes $\mathcal{W}[1]$ and $\mathcal{W}[2]$. Proving hardness with respect to these classes can be done using a *parameterized reduction*, which reduces a problem instance (I, k) in $f(k) \cdot |I|^{O(1)}$ time to an instance (I', k') such that (I, k) is a yes-instance if and only if (I', k') is a yes-instance, and k' only depends on k but not on $|I|$, see [8, 15, 18].

For all our hardness proofs, we use the $\mathcal{W}[2]$-complete DOMINATING SET (DS) problem for undirected graphs.

DOMINATING SET
Given: An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and a nonnegative integer k.
Question: Is there a subset $\mathcal{V}' \subseteq \mathcal{V}$ with $|\mathcal{V}'| \leq k$ such that every vertex $v \in \mathcal{V}$ is contained in $\mathcal{V}'$ or has a neighbor in $\mathcal{V}'$?

Notation in our proofs. In all our reductions from DOMINATING SET, we will associate the vertices of the given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with candidates of the election $E = (C, V)$ to be constructed. For that sake, we use a bijection $g \colon \mathcal{V} \to C$. By $N(v) := \{u \in \mathcal{V} \mid \{u, v\} \in \mathcal{E}\}$, we denote the set of *neighbors* or the *neighborhood* of a vertex $v \in \mathcal{V}$. Analogously, we define the *neighborhood of a candidate* c_i as $N(c_i) = g(N(v_i))$ for $c_i = g(v_i)$, i.e., the set of neighbors of a candidate $c_i \in C$ corresponding to the vertex $v_i \in \mathcal{V}$ is the set of candidates corresponding to the neighborhood of v_i in $\mathcal{G}$. By $\overline{N(v_i)}$ we denote the set of non-neighbors of v_i, analogously for neighborhoods of candidates.

In our reductions, we usually need one dummy candidate for every $c_i \in C$, these will be denoted by $\hat{c}_i$. All other dummy candidates appearing are marked with a hat as well, usually they are called $\hat{d}$ or similarly. When building the votes in our reductions, we write '$k \parallel a_1 \succ \cdots \succ a_l$' which means that we construct the given vote $a_1 \succ \cdots \succ a_l$ exactly k times.

In our preference lists, we sometimes specify a whole subset of candidates, e.g., $c \succ D$ for a candidate $c \in C$ and a subset of candidates $D \subseteq C$. This notation means $c \succ d_1 \succ \cdots \succ d_l$

for an arbitrary but fixed order of $D = \{d_1, \ldots, d_l\}$. If we use a set $\overrightarrow{D}$ in a preference list, we mean one specific, fixed (but arbitrary, and unimportant) order of the elements in D, which is reversed if we write $\overleftarrow{D}$. Hence, if $c \succ \overrightarrow{D}$ stands for $c \succ d_1 \succ \cdots \succ d_l$, then $c \succ \overleftarrow{D}$ means $c \succ d_l \succ \cdots \succ d_1$. Finally, whenever we use the notation D_{rest} for a subset of candidates in a vote, we mean the set consisting of those candidates in D that have not been positioned explicitly in this vote.

3 Complexity-theoretic analysis

In this section, we will give several hardness proofs for Voter Deterrence for different voting systems. All our results rely on reductions from the $\mathcal{NP}$-complete problem Dominating Set. We only prove $\mathcal{NP}$-hardness for the different voting systems, but since membership in $\mathcal{NP}$ is always trivially given, $\mathcal{NP}$-completeness follows immediately. For all these reductions we assume that every vertex of the input instance has at least two neighbors, which is achievable by a simple polynomial time preprocessing.

3.1 Plurality

It is easy to see that 1-Voter Deterrence can be solved efficiently for Plurality. One can simply order the candidates according to their score and if there are more than k candidates ahead of p, this instance is a no-instance. Otherwise p will win after deletion of the candidates that were ranked higher than him, because all the votes which they got a point from are removed. Therefore the following theorem holds.

Theorem 1. 1-Voter Deterrence *is in* $\mathcal{P}$ *for the voting system* Plurality.

For 2-Voter Deterrence, it is not so easy to see which candidates should be deleted. In fact, the problem is $\mathcal{NP}$-complete.

Theorem 2. 2-Voter Deterrence *is* $\mathcal{NP}$*-complete for the voting system* Plurality.

Proof. We prove Theorem 2 with a parameterized reduction from Dominating Set. Let $\langle \mathcal{G} = (\mathcal{V}, \mathcal{E}), k \rangle$ be an instance of DS.
Candidates: For every vertex $v_i \in \mathcal{V}$ we need one candidate c_i and one dummy candidate $\hat{c}_i$, as well as the preferred candidate p and his dummy candidate $\hat{p}$, so $C = I \cup D \cup \{p\}$ with $I = \{c_1, \ldots, c_n\}$ and $D = \{\hat{c}_1, \ldots, \hat{c}_n, \hat{p}\}$. For ease of presentation we denote $I \cup \{p\}$ by I^*.
Votes: The votes are built as follows.

$$
\begin{aligned}
& n \parallel p \succ \hat{p} \succ C_{\text{rest}}, && (1)\\
& \forall c_i \in I: \\
& \quad |N(c_i)| \parallel c_i \succ \hat{c}_i \succ C_{\text{rest}}, && (2)\\
& \quad \forall c_j \in \ I^* \setminus (N(c_i) \cup \{c_i\}): \\
& \qquad 1 \parallel c_i \succ c_j \succ C_{\text{rest}}. && (3)
\end{aligned}
$$

Note that n votes are built for every candidate c_i. Therefore each candidate in I^* has the score n. The score of a candidate can only be decreased if the corresponding candidate himself is deleted. Note also that the score of every dummy candidate cannot exceed $n-1$. We will now show that one can make p win the election by deleting up to k candidates if and only if the DS-instance has a solution of size at most k.

"⇒": Let S be a given solution to the DS-instance. Then $R = g(S)$ is a solution to the corresponding 2-Voter Deterrence-instance. Since S is a dominating set, every candidate in I will be at least once in the neighborhood of a candidate $c_i \in R$ or be a

candidate in R himself. Therefore p is the only candidate who gains an additional point from every deleted candidate $c_x \in R$ from the vote built by (3) and will therefore be the unique winner.

"⇐": Let R be a given solution to a 2-Voter Deterrence-instance. Since every candidate in I^* has the original score n and these scores can only be increased if the corresponding candidate himself is not deleted, as discussed before, every candidate $c_x \in I$ must not appear as c_j on the second position of the votes built by (3) for at least one candidate of R or be a member of R himself. Therefore $S = g^{-1}(R)$ is a solution to the equivalent DS-instance. □

3.2 Veto

Theorem 3. 1-Voter Deterrence *is $\mathcal{NP}$-complete for the voting system* Veto.

Proof. We prove Theorem 3 with a parameterized reduction from Dominating Set. Let $\langle \mathcal{G} = (\mathcal{V}, \mathcal{E}), k \rangle$ be an instance of DS.
Candidates: For every vertex $v_i \in \mathcal{V}$ we need one candidate c_i, as well as the preferred candidate p and $k+1$ dummy candidates, so $C = I \cup D \cup \{p\}$ with $I = \{c_1, \ldots, c_n\}$ and $D = \{\hat{d}_1, \ldots, \hat{d}_{k+1}\}$. For ease of presentation we denote $I \cup \{p\}$ by I^*.
Votes: The votes are built as follows.

$$\begin{aligned}
&\forall c_i \in I: \\
&\quad \forall c_j \in I^* \setminus (N(c_i) \cup \{c_i\}): \\
&\qquad 1 \parallel c_i \succ C_{\text{rest}} \succ D \succ c_j, && (1)\\
&\quad \forall c_j \in N(c_i) \cup \{c_i\}: \\
&\qquad 1 \parallel p \succ I_{\text{rest}} \succ D \succ c_j, && (2)\\
&\quad \forall \hat{d}_j \in D: \\
&\qquad 2 \parallel p \succ I \succ D_{\text{rest}} \succ \hat{d}_j. && (3)
\end{aligned}$$

Note that every vote built by (2) and (3) can only be removed by deleting the candidate p, who should win the election. Therefore these votes will not be removed. Note also that for each set of votes constructed for a candidate $c_i \in I$, every candidate in $C \setminus D$ takes the last position in one of theses votes, hence the score of every such candidate is the same. In contrast, the dummy candidates cannot win the election at all, due to the fact that they are on the last position of the constructed votes twice as often as the other candidates.
We will now show that one can make p win the election by deleting up to k candidates if and only if the DS-instance has a solution of size at most k.

"⇒": Let S be a given solution to the DS-instance. Then $R = g(S)$ is a solution to the corresponding 1-Voter Deterrence-instance. Since S is a dominating set, every candidate in I will be on the last position of a vote built by (2) for a $c_j \in R$ at least once and therefore lose a point relative to p, hence p is the unique winner.

"⇐": Let R be a given solution to a 1-Voter Deterrence-instance. As discussed before, only votes built by (1) can be removed by deleting a candidate. Since at most k candidates can be deleted, it is not helpful to delete a dummy candidate, because they have less points than p and their deletion cannot decrease the points of any candidate in I (which are actually holding p from winning). Therefore only candidates in I are in R, or there exists a solution $R' \subseteq R$, for which this holds. With every candidate chosen from I, the corresponding neighbors are losing one point relative to p. As p and every candidate of I had the same amount of points in the beginning, every candidate in I has to be at least neighboring one deleted candidate or be deleted himself. By the definition of the neighborhood of candidates, $S = g^{-1}(R')$ is a solution to the equivalent DS-instance. □

3.3 2-approval

Theorem 4. 1-Voter Deterrence *is* $\mathcal{NP}$*-complete for the voting system* 2-approval.

Proof. We prove Theorem 4 by a parameterized reduction from Dominating Set. Let $\langle \mathcal{G} = (\mathcal{V}, \mathcal{E}), k \rangle$ be an instance of DS.
Candidates: For every vertex $v_i \in \mathcal{V}$, we create one candidate c_i and one additional dummy candidate $\hat{c}_i$, finally we need the preferred candidate p. So with $I = \{c_1, \ldots, c_n\}$ and $D = \{\hat{c}_1, \ldots, \hat{c}_n\}$, the candidates are $C = I \cup D \cup \{p\}$.
Votes: The votes are built as follows.

$$\begin{aligned}
&\forall c_i \in I: \\
&\quad \forall c_j \in N(c_i): \\
&\qquad 1 \parallel c_i \succ c_j \succ \hat{c}_j \succ C_{\text{rest}} \succ p, && (1)\\
&\quad \forall c_j \in I \setminus (N(c_i) \cup \{c_i\}): \\
&\qquad 1 \parallel \hat{c}_i \succ c_j \succ \hat{c}_j \succ C_{\text{rest}} \succ p, && (2)\\
&\quad 2 \parallel \hat{c}_i \succ p \succ C_{\text{rest}}, && (3)\\
&\quad n - |N(c_i)| \parallel c_i \succ \hat{c}_i \succ C_{\text{rest}} \succ p. && (4)
\end{aligned}$$

Without any candidate deleted, all $c_i \in I$ and p have the same score of $2n$, while the dummy candidates $\hat{c}_j \in D$ have a score less than $2n$. Note that one decreases p's score by deleting a dummy candidate, because a deletion of this kind results in losing a vote built in (3). Therefore one has to delete candidates in I to help p in winning.
We will now show that one can make p win the election by deleting up to k candidates if and only if the DS-instance has a solution of size at most k.

"⇒": Let S be a given solution to the DS-instance. Then $R = g(S)$ is a solution to the corresponding 1-Voter Deterrence-instance. Since S is a dominating set, every candidate $c_x \in I$ will be at the second position of a vote built by (1) for one $c_i \in R$ at least once and therefore lose a point. As a consequence, every corresponding dummy candidate $\hat{c}_x$ will have a score not greater than $2n - 2$, as they gain points in votes built by (1) and (2), by succeeding to position 2, but lose points as a result of the removal of votes built by (4). Consequently, p wins being the only candidate remaining with a score of $2n$.

"⇐": Let R be a given solution to a 1-Voter Deterrence-instance. Since one cannot increase p's score by deleting a candidate $c_i \in I$, the deletion of the candidates in R has to reduce the scores of all candidates in I by at least 1. Whenever a dummy candidate is deleted, p loses points instead of gaining them, therefore $R \subseteq I$ must hold. To reduce the score of every candidate in I by just deleting candidates in I, every such candidate has to be in the neighborhood of at least one deleted candidate or be deleted himself. By the definition of the neighborhood of candidates, $S = g^{-1}(R)$ is a solution to the equivalent DS-instance. □

3.4 Borda

Theorem 5. 1-Voter Deterrence *is* $\mathcal{NP}$*-complete for the voting system* Borda.

Proof. We prove Theorem 5 by a parameterized reduction from Dominating Set. Let $\langle \mathcal{G} = (\mathcal{V}, \mathcal{E}), k \rangle$ be an instance of DS.
Candidates: For every vertex $v_i \in \mathcal{V}$ we create one candidate c_i and one dummy candidate $\hat{c}_i$, finally we need the preferred candidate p. So the candidates are $C = I \cup D \cup \{p\}$ with $I = \{c_1, \ldots, c_n\}$ and $D = \{\hat{c}_1, \ldots, \hat{c}_n\}$. For ease of presentation, we denote $I \cup \{p\}$ by I^*.

Votes: The votes are built as follows.

$$\forall c_i \in I:$$
$$\forall c_j \in N(c_i):$$
$$1 \parallel c_i \succ \overrightarrow{I^*}_{\text{rest}} \succ c_j \succ \hat{c}_j \succ \overrightarrow{D}_{\text{rest}} \succ \hat{c}_i, \quad (1)$$
$$1 \parallel c_i \succ c_j \succ \hat{c}_j \succ \overleftarrow{I^*}_{\text{rest}} \succ \overleftarrow{D}_{\text{rest}} \succ \hat{c}_i, \quad (2)$$
$$1 \parallel \hat{c}_i \succ \hat{c}_j \succ c_j \succ \overleftarrow{I^*}_{\text{rest}} \succ c_i \succ \overleftarrow{D}_{\text{rest}}, \quad (3)$$
$$1 \parallel \hat{c}_i \succ \overrightarrow{I^*}_{\text{rest}} \succ \hat{c}_j \succ c_j \succ c_i \succ \overrightarrow{D}_{\text{rest}}. \quad (4)$$

Recall that $\overrightarrow{A}$ denotes one specific order of the elements within the set A which is reversed in $\overleftarrow{A}$. Keeping this in mind, it is easy to see that every candidate in I^* has the same score within one gadget constructed by the four votes built by (1) to (4) for one c_j, while the dummy candidates all have a lower score. Note that the deletion of any candidate will decrease the score of every other candidate. Therefore the scores of the candidates in I have to be decreased more than the one of p, whereas the scores of the candidates in I^* can never be brought below the score of any candidate in D.

We will now show that one can make p win the election by deleting up to k candidates if and only if the DS-instance has a solution of size at most k.

"⇒": Let S be a given solution to the DS-instance. Then $R = g(S)$ is a solution to the corresponding 1-Voter Deterrence-instance. Since S is a dominating set, every candidate $c_x \in I$ will appear at least once as c_j in the votes built by (1) to (4) for one $c_i \in R$ and therefore lose two points relative to p. With the dummy candidates unable to reach a higher score than p and every other candidate having a score below the one of p, the preferred candidate wins.

"⇐": Let R be a given solution to a 1-Voter Deterrence-instance. Since one cannot increase the score of p, the deletion of the candidates in R has to decrease the score of every candidate of I relative to p. Therefore every candidate in I has to appear at least once as c_j in the votes built by (1) to (4) for one $c_i \in R$. Hence, every candidate of I must have at least one neighbor in R or be a member of R himself. Therefore $S = g^{-1}(R)$ is a solution to the equivalent DS-instance. □

3.5 Maximin

Theorem 6. 1-Voter Deterrence *is $\mathcal{NP}$-complete for the voting system* Maximin.

Proof. We prove Theorem 6 by a parameterized reduction from Dominating Set. Let $\langle \mathcal{G} = (\mathcal{V}, \mathcal{E}), k\rangle$ be an instance of DS.

Candidates: For every vertex $v_i \in \mathcal{V}$ we create one candidate c_i and one dummy candidate $\hat{c}_i$, finally we need the preferred candidate p. So the candidates are $C = I \cup D \cup \{p\}$ with $I = \{c_1, \dots, c_n\}$ and $D = \{\hat{c}_1, \dots, \hat{c}_n\}$.

Votes: The votes are built as follows.

$$\forall c_i \in I:$$
$$1 \parallel c_i \succ \overrightarrow{I}_{\text{rest}} \succ \overrightarrow{N}(c_i) \succ p \succ \overrightarrow{D}_{\text{rest}} \succ \hat{c}_i, \quad (1)$$
$$1 \parallel c_i \succ \overleftarrow{N}(c_i) \succ p \succ \overleftarrow{I}_{\text{rest}} \succ \overleftarrow{D}_{\text{rest}} \succ \hat{c}_i, \quad (2)$$
$$1 \parallel \hat{c}_i \succ \overrightarrow{I}_{\text{rest}} \succ p \succ \overrightarrow{N}(c_i) \succ \overrightarrow{D}_{\text{rest}} \succ c_i, \quad (3)$$
$$1 \parallel \hat{c}_i \succ p \succ \overleftarrow{N}(c_i) \succ \overleftarrow{I}_{\text{rest}} \succ \overleftarrow{D}_{\text{rest}} \succ c_i. \quad (4)$$

Recall that $\overrightarrow{A}$ denotes one specific order of the elements within set A which is reversed in $\overleftarrow{A}$. With this in mind, it is easy to see that every candidate in I has the same score as p,

namely $2n$. The dummy candidates are not able to win the election as long as at least one of the candidates in I or p is remaining.
We will now show that one can make p win the election by deleting up to k candidates if and only if the DS-instance has a solution of size at most k.

"⇒": Let S be a given solution to the DS-instance with $|S| = k' \leq k$. Then $R = g(S)$ is a solution to the corresponding 1-Voter Deterrence-instance. Since S is a dominating set, every candidate $c_x \in I$ will belong to the neighborhood of a candidate in R or be a member of R himself at least once. Therefore each candidate c_x will have at most $2n - k' - 2$ votes in which he is preferred to p. Therefore the maximin score of these candidates will be at most $2n - k' - 2$, while p is preferred to every other candidate in C in at least $2n - k'$ votes, which makes p the unique winner of the election.

"⇐": Let R be a given solution to a 1-Voter Deterrence-instance. Then the deletion of the candidates in R decreases the score of every candidate in I more than the score of p. Note that the score of p is always higher than the score of the dummy candidates. The only way to decrease the score of a candidate $c_x \in I$ is to delete c_x himself, or one of his neighbors, since this removes the votes built by (1) and (2), in which the neighbors are preferred to p, while p is preferred in the remaining votes built by (3) and (4). Since every candidate has to be in the neighborhood of at least one deleted candidate or be deleted himself, $S = g^{-1}(R)$ is a solution to the equivalent DS-instance. □

3.6 Bucklin and Fallback Voting

A candidate c's Bucklin score is the smallest number k such that more than half of the votes rank c among the top k candidates. The winner is the candidate that has the smallest Bucklin score [20].

Theorem 7. 1-Voter Deterrence *is $\mathcal{NP}$-complete for* Bucklin.

Note that Bucklin is a special case of *Fallback Voting*, where each voter approves of each candidate, see [9]. We therefore also obtain

Corollary 1. 1-Voter Deterrence *is $\mathcal{NP}$-complete for* Fallback Voting.

Proof. We prove Theorem 7 by a parameterized reduction from Dominating Set. Let $\langle \mathcal{G} = (\mathcal{V}, \mathcal{E}), k \rangle$ be an instance of DS.
Candidates: For every vertex $v_i \in \mathcal{V}$ we create one candidate c_i and one dummy candidate $\hat{c}_i$. Additionally, we need the preferred candidate p and several dummy candidates. We need $n(n+k)$ *filling* dummies $\hat{f}$, $k(2n+k-1)$ *security* dummies $\hat{s}$, and finally $k-1$ *leading* dummies $\hat{l}$. So the candidates are $C = I \cup D \cup S \cup F \cup L \cup \{p\}$ with $I = \{c_1, \ldots, c_n\}$, $D = \{\hat{c}_1, \ldots, \hat{c}_n\}$, $S = \{\hat{s}_1, \ldots, \hat{s}_{k(2n+k-1)}\}$, $F = \{\hat{f}_1, \ldots, \hat{f}_{n(n+k)}\}$, and $L = \{\hat{l}_1, \ldots, \hat{l}_{k-1}\}$. For ease of presentation, we denote $I \cup \{p\}$ by I^*.
Votes: The votes are built as follows.

$\forall c_i \in I:$

$$1 \parallel c_i \succ N(c_i) \succ \{\hat{f}_{(i-1)(n+1)+1}, \ldots, \hat{f}_{i(n+1)-|N(c_i)|-1}\} \\ \succ \{\hat{s}_{(2i-2)(k+1)+1}, \ldots, \hat{s}_{2(i-1)(k+1)}\} \succ C_{\text{rest}} \succ p, \quad (1)$$

$$1 \parallel \hat{c}_i \succ \overline{N(c_i)} \succ \{\hat{f}_{i(n+1)-|N(c_i)|}, \ldots, \hat{f}_{(i)(n+1)}\} \succ p \\ \succ \{\hat{s}_{(2i-1)(k+1)+1}, \ldots, \hat{s}_{2i(k+1)}\} \succ C_{\text{rest}}, \quad (2)$$

$\forall r \in \{1, \ldots, k-1\}:$ one vote of the form

$$1 \parallel \hat{l}_r \succ \{\hat{f}_{n(n+1)+(r-1)n+1}, \ldots, \hat{f}_{n(n+1)+in}\} \\ \succ \{\hat{s}_{2n(k+1)+(r-1)(k+1)+1}, \ldots, \hat{s}_{2n(k+1)+r(k+1)}\} \succ C_{\text{rest}} \succ p. \quad (3)$$

Note that every candidate in I^* occurs within the first $n+2$ positions in the votes built by (1) and (2) for every candidate $c_i \in I$ exactly once. Therefore p is not the unique winner without modification. Note also that deleting some of the dummy candidates is not helping p, as they all appear just once within the first $n+2$ positions. Because of the security dummies, no candidate in I^* can move up to one of the first $n+2$ positions, if he has not been there before. After the deletion of k candidates, up to k votes can be removed—note that every removed vote has to be built by (1) or (3) if p wins the election with this deletion. We will now show that one can make p win the election by deleting up to k candidates if and only if the DS-instance has a solution of size at most k.

"⇒": Let S be a given solution to a DS-instance with $|S| = k' \leq k$. Then $R = g(S) \cup \{\hat{l}_1, \ldots, \hat{l}_{k-k'}\}$ is a solution to the corresponding 1-Voter Deterrence-instance. Since S is a dominating set, every candidate $c_x \in I$ will lose at least one vote built by (1) because he is the neighbor of at least one candidate in R or a member of R himself. Since $|R| = k$, k votes are removed and therefore the score of p is $n+2$, whereas the score of every other candidate is greater than $n+2$, which makes p win the election.

"⇐": Let R be a given solution to a 1-Voter Deterrence-instance. Since p wins with the candidates in R deleted, R has to contain just candidates in $I \cup L$, and $|R| = k$, because everything else would increase the score of p to the maximum, which would keep p from winning uniquely. Let $R' = R \cap I$ be the intersection of R and I. Since the score of p with k removed votes of this kind is $n+2$, and the score of every candidate in I was $n+2$ without the removal of any votes, every candidate in I has to be removed himself or has to be neighboring at least one deleted candidate in R, because only then his score is greater than $n+2$. Therefore $S = g^{-1}(R')$ is a solution to the equivalent DS-instance. □

3.7 Copeland

For any two distinct candidates i and j, let $N(i,j)$ be the number of voters that prefer i to j, and let $C(i,j) = +1$ if $N(i,j) > N(j,i)$, $C(i,j) = 0$ if $N(i,j) = N(j,i)$, and $C(i,j) = -1$ if $N(i,j) < N(j,i)$. The *Copeland score* of candidate i is $\sum_{j \neq i} C(i,j)$ [6].

Theorem 8. 1-Voter Deterrence *is $\mathcal{NP}$-complete for the voting system* Copeland.

Proof. We prove Theorem 8 by a parameterized reduction from Dominating Set. Let $\langle \mathcal{G} = (\mathcal{V}, \mathcal{E}), k \rangle$ be an instance of DS.
Candidates: For every vertex $v_i \in \mathcal{V}$ we create one candidate c_i and one dummy candidate $\hat{c}_i$. Additionally we need the preferred candidate p, one *thievish* candidate $\hat{t}$ and furthermore n *filling* dummy candidates. So the candidates are $C = I \cup D \cup F \cup \{\hat{t}, p\}$ with $I = \{c_1, \ldots, c_n\}$, $D = \{\hat{c}_1, \ldots, \hat{c}_n\}$, and $F = \{\hat{f}_1, \ldots, \hat{f}_n\}$.
Votes: The votes are built as follows.

$$\forall c_i \in I:$$
$$1 \parallel c_i \succ \overrightarrow{N}(c_i) \succ \hat{t} \succ \overrightarrow{I}_{\text{rest}} \succ p \succ \overrightarrow{F} \succ \overrightarrow{D}_{\text{rest}} \succ \hat{c}_i, \quad (1)$$
$$1 \parallel c_i \succ p \succ \overleftarrow{I}_{\text{rest}} \succ \overleftarrow{N}(c_i) \succ \overleftarrow{F} \succ \hat{t} \succ \overleftarrow{D}_{\text{rest}} \succ \hat{c}_i, \quad (2)$$
$$1 \parallel \hat{c}_i \succ \hat{t} \succ \overrightarrow{N}(c_i) \succ \overrightarrow{I}_{\text{rest}} \succ p \succ \overrightarrow{F} \succ c_i \succ \overrightarrow{D}_{\text{rest}}, \quad (3)$$
$$1 \parallel \hat{c}_i \succ p \succ \overleftarrow{I}_{\text{rest}} \succ \overleftarrow{F} \succ \hat{t} \succ \overleftarrow{N}(c_i) \succ c_i \succ \overleftarrow{D}_{\text{rest}}. \quad (4)$$

After creating these n gadgets (consisting of the above 4 votes) the candidates have different scores. Note that the candidates of each set are always tying with the other candidates in their set, since every gadget has two votes with one specific order of the members and another two of the reversed order. Since candidates in D are losing every pairwise election against all other candidates, they have a score of $-(2n+2)$. The candidates

in F are just winning against the candidates in D and are tied against $\hat{t}$ and therefore have a score of -1. Since the candidates in I and p are on a par with $\hat{t}$, this gives them a score of $2n$ and $\hat{t}$ a score of n. Note that if there exists a deletion of k candidates which makes p win the election, there also exists a deletion of up to k candidates in I doing so. The main idea here is that the thievish candidate can steal exactly one point from every candidate in I by winning the pairwise election between them due to the deleted candidate and thereby removed votes. Since $\hat{t}$ starts with a score of n, this will only bring him to a score of $2n - k$ with k deleted candidates. Therefore he cannot get a higher score than p initially had.
We will now show that one can make p win the election by deleting up to k candidates if and only if the DS-instance has a solution of size at most k.

"$\Rightarrow$": Let S be a given solution to a DS-instance. Then $R = g(S)$ is a solution to the corresponding 1-Voter Deterrence-instance. Since S is a dominating set, every candidate $c_x \in I$ will be a neighbor of a deleted candidate, or a deleted candidate himself. Therefore $\hat{t}$ will win the pairwise election with every such candidate c_x due to the fact that initially they were tied, but at least one vote built by (1) and one by (2) are deleted, where c_x was in the neighborhood of the deleted c_i, or c_x got deleted himself. As a consequence, $\hat{t}$ has a score of $2n - k$ and every candidate in I has a score of $2n - 1$, which makes p win the election with an unchanged score of $2n$.

"$\Leftarrow$": Let R be a given solution to a 1-Voter Deterrence-instance. As discussed before, there must be a solution R' of size at most k with $R' \cap (C \setminus I) = \emptyset$. Since p and the candidates in I were leading initially with the same score of $2n$, and p cannot get a higher score if any candidate is deleted, the candidates in I must have their score lowered through deletion of some candidates. Any deleted candidate himself cannot win anymore, but since only up to k candidates are to delete, the remaining candidates in I have to lose at least one pairwise election after the deletion, which they won or at least tied before. By design of the gadget, this can only be achieved for a candidate c_x by deleting c_i with $c_x \in N(c_i)$. This makes c_x lose the former tied pairwise election with $\hat{t}$, giving c_x a score of $2n-1$. Since this must hold for every candidate in I and therefore any non-deleted candidate must be a neighbor of one candidate in M' at least. Hence $S = g^{-1}(R')$ is a solution to the equivalent DS-instance. □

4 Parameterized complexity-theoretic analysis

In this section, we shortly take a closer look at the parameterized complexity of Voter Deterrence for the previously considered voting systems.

Since all the $\mathcal{NP}$-hardness proofs of the previous section are based on parameterized reductions from Dominating Set, we immediately obtain

Corollary 2. *1-Voter Deterrence is $\mathcal{W}[2]$-hard for* Copeland*,* Veto*,* Borda*,* 2-approval*,* Maximin*,* Bucklin*, and* Fallback Voting*, and 2-Voter Deterrence is $\mathcal{W}[2]$-hard for* Plurality*, all with respect to the parameter* number of deleted candidates*.*

In contrast, considering a different parameter, one easily obtains the following tractability result.

Theorem 9. *The problem* x-Voter Deterrence *is in $\mathcal{FPT}$ with respect to the parameter* number of candidates *for all voting systems having a polynomial time winner determination.*

Proof. It is easy to see that Theorem 9 holds: An algorithm trying out every combination of candidates to delete has an $\mathcal{FPT}$-running time $\mathcal{O}(m^k \cdot n \cdot m \cdot T_{\text{poly}})$, where m is the number of candidates, n the number of votes, $k \leq m$ is the number of allowed deletions, and T_{poly} is the polynomial running time of the winner determination in the specific voting system. □

5 Conclusion

We have initiated the study of a voting problem that takes into account correlations that appear in real life, but which has not been considered from a computational point of view so far. We obtained $\mathcal{NP}$-completeness and $\mathcal{W}[2]$-hardness for most voting systems we considered. However, this is just the beginning, and it would be interesting to obtain results for other voting systems such as k-approval or scoring rules in general. Also, we have concentrated on the case of 1-Voter Deterrence and so far have investigated 2-Voter Deterrence for Plurality only.

One could also look at the *destructive* variant of the problem in which an external agent wants to prevent a hated candidate from winning the election, see e.g. [17] for a discussion for the 'control' problems.

We have also investigated our problem from the point of view of parameterized complexity. It would be interesting to consider different parameters, such as the number of votes, or even a combination of several parameters (see [19]), to determine the complexity of the problem in a more fine-grained way. This approach seems especially worthwile because Voter Deterrence, like other ways of manipulating the outcome of an election, is a problem for which NP-hardness results promise some kind of resistance against this dishonest behavior. Parameterized complexity helps to keep up this resistance or to show its failure for cases where certain parts of the input are small, and thus provides a more robust notion of hardness. See, e.g., [3–5,7,9], and the recent survey [2].

However, one should keep in mind that combinatorial hardness is a worst case concept, so it would clearly be interesting to consider the average case complexity of the problem or to investigate the structure of naturally appearing instances. E.g., when the voters have *single peaked preferences*, many problems become easy [13]. Research in this direction is becoming more and more popular, see for example [13, 14, 16].

Acknowledgments. We are grateful to the anonymous referees whose constructive feedback helped to improve this work. Finally, we thank Oliver Gableske for the fruitful discussion which initiated our study of Voter Deterrence.

References

[1] J. Bartholdi, C. Tovey, M. Trick, et al. How Hard is it to Control an Election? *Mathematical and Computer Modelling*, 16(8-9):27–40, 1992.

[2] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. Studies in Computational Aspects of Voting—a Parameterized Complexity Perspective. In H. Bodlaender et al., editor, *Fellows Festschrift,* LNCS 7370, pages 318–363. Springer, Heidelberg, 2012.

[3] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 410(52):5425–5442, 2009.

[4] R. Bredereck, J. Chen, S. Hartung, R. Niedermeier, O. Suchỳ, and S. Kratsch. A multivariate complexity analysis of lobbying in multiple referenda. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI '12)*, 2012. Accepted for publication.

[5] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.

[6] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? *CoRR*, cs.GT/0307003, 2003.

[7] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 2011. Available electronically.

[8] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer, New York, 1999.

[9] G. Erdélyi and M. Fellows. Parameterized control complexity in bucklin voting and in fallback voting. *Proceedings of COMSOC*, 10, 2010.

[10] G. Erdélyi and J. Rothe. Control complexity in fallback voting. In *Proceedings of the Sixteenth Symposium on Computing: the Australasian Theory - Volume 109*, CATS '10, pages 39–48, Darlinghurst, Australia, Australia, 2010. Australian Computer Society, Inc.

[11] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305, 2011.

[12] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35(1):275–341, 2009.

[13] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Inf. Comput.*, 209(2):89–107, 2011.

[14] P. Faliszewski and A. D. Procaccia. Ai's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[15] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, New York, 2006.

[16] E. Friedgut, G. Kalai, and N. Nisan. Elections can be manipulated often. In *FOCS*, pages 243–249. IEEE Computer Society, 2008.

[17] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artif. Intell.*, 171(5-6):255–285, 2007.

[18] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, 2006.

[19] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *STACS 2010: Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science*, pages 17–32, 2010.

[20] L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proc. 21st IJCAI*, pages 348–353, 2009.

Britta Dorn
Wilhelm-Schickard-Institut für Informatik
Sand 13
Universität Tübingen
72076 Tübingen, Germany
Email: `britta.dorn@uni-tuebingen.de`

Dominikus Krüger
Institut für Theoretische Informatik
James-Franck-Ring 5 / O27
Universität Ulm
89081 Ulm, Germany
Email: `dominikus.krueger@uni-ulm.de`

Graph Aggregation

Ulle Endriss and Umberto Grandi

Abstract

Suppose a number of agents each provide us with a directed graph over a common set of vertices. Graph aggregation is the problem of computing a single graph that best represents the information inherent in this profile of individual graphs. We introduce a simple formal framework for graph aggregation and then focus on the notion of collective rationality, which asks whether a given property of graphs, such as transitivity, can be guaranteed to hold for the collective graph whenever it is satisfied by all individual graphs. We refine the ultrafilter method for proving impossibility theorems in social choice theory to arrive at a clear picture relating axiomatic properties of aggregation procedures, properties of graphs with respect to which we want to ensure collective rationality, and properties of ultrafilters.

1 Introduction

Suppose a group of agents each supply us with a particular piece of information and we want to aggregate this information into a collective view to obtain a good representation of the individual views provided. In classical social choice theory the objects of aggregation have been preference orders on a set of alternatives (Arrow, 1963). More recently, the same methodology has also been applied to other types of information, notably beliefs (Konieczny and Pino Pérez, 2002), judgments (List and Puppe, 2009), ontologies (Porello and Endriss, 2011), and rankings provided by Internet search engines (Dwork et al., 2001).

In this paper, we introduce the problem of *graph aggregation*, i.e., the problem of devising methods to aggregate the information inherent in a profile of individual (directed) graphs, one for each agent, into a single collective graph. Given that a preference order is a special kind of directed graph, graph aggregation may be viewed as a direct generalisation of classical preference aggregation. This is a useful generalisation, because also several other problem domains in which aggregation is relevant are naturally modelled as graphs, e.g.:

- In abstract argumentation (Dung, 1995), collections of arguments available for a debate are modelled as a graph (with an edge from A to B if argument A *attacks* B). The question of how to integrate several such argument graphs naturally arises in this context. Recent work of Coste-Marquis et al. (2007) has addressed this question.
- Social and economic networks are often modelled as graphs (Jackson, 2008).[1] We might want to merge the information from several such networks (e.g., the network of work relations in a community, the network of friends in the same community, etc.).
- It is not always reasonable to take the classical assumptions of economic theory (according to which preferences are transitive and complete orders) for granted when modelling an agent's preferences. The work of Pini et al. (2009) goes in this direction by studying aggregation of preferences modelled as incomplete orders; but we might want to go further and also allow for cycles and so forth.

Special instances of the graph aggregation problem we shall consider have previously been studied in work on the aggregation of judgments regarding causal relations between variables

[1] While social networks are usually modelled as undirected graphs, here we shall work with directed graphs (but note that we can model an undirected graph as a directed graph that is symmetric).

(Bradley et al., 2011) and the design of voting agendas for multi-issue elections based on individually reported preferential dependencies between issues (Airiau et al., 2011).

While graph aggregation is more general than preference aggregation, it is less general than the frameworks of judgment aggregation (List and Puppe, 2009) or binary aggregation (Dokow and Holzman, 2010; Grandi and Endriss, 2010): just like classical preference aggregation, graph aggregation can—in principle—be embedded into these frameworks. For a given problem domain, it is important to find the right level of abstraction, and graphs appear to be a particularly useful level of abstraction for a wide range of problems.

In Section 2, we define a framework for graph aggregation and adapt well-known axioms from the literature to express natural desiderata for such aggregators. We also suggest concrete aggregators, including both adaptations from other areas of social choice theory and the novel class of *successor-approval rules.*

Our main interest will then be in the notion of *collective rationality.* In Section 3, we define collective rationality of a given aggregator F wrt. a given property of graphs P (such as transitivity) as the guarantee that, whenever each of the individual graphs satisfies P, so does the collective graph we obtain when we apply F to those individual graphs. That is, assuming that each individual agent is "rational" in the sense of respecting the property P under consideration, we ask whether we can be sure that the collective (as defined by our aggregator F) will be rational as well. This is a well-known concept: in classical preference aggregation, P corresponds to the conjunction of the properties that define a weak order (Arrow, 1963); in judgment aggregation, P corresponds to logical consistency (List and Puppe, 2009); and in our own previous work on binary aggregation, P corresponds to an integrity constraint expressed in a propositional language (Grandi and Endriss, 2010).

We first prove a series of simple results that identify certain (classes of) aggregators that are collectively rational wrt. certain properties of graphs. Our main technical contribution is a refinement of the *ultrafilter method* for proving impossibility theorems in social choice theory (see, e.g., Kirman and Sondermann, 1972; Herzberg and Eckert, 2011). One way of proving Arrow's classical impossibility theorem (Arrow, 1963) is to show that the collection of winning coalitions of individuals (determining which pieces of information need to be accepted by an aggregator) is an ultrafilter (Davey and Priestley, 2002). We will show how each of the conditions defining an ultrafilter corresponds directly to the requirement of collective rationality wrt. a certain graph property. For example, any property that, given the acceptance of two particular edges, forces the acceptance of a third edge can be used to establish the ultrafilter condition of being closed under intersections. This means that we can replace transitivity in the statement of an Arrow-like theorem by, say, the Church-Rosser property or the Euclidean property (see Table 1 for definitions of these three properties, all of which have the general template indicated before). We use our technique to prove several variants of Arrow's Theorem for graph aggregation.

Section 4 concludes with a discussion of related research and directions for future work.

2 A Formal Framework of Graph Aggregation

Fix a finite set of *vertices* V. A (directed) *graph* $G = \langle V, E \rangle$ based on V is defined by a set of *edges* $E \subseteq V^2$. Let $\mathcal{G}$ be the set of all such graphs (for our fixed choice of V). Let $\mathcal{N}$ be a finite set of (two or more) *individuals* (or *agents*). Each individual $i \in \mathcal{N}$ provides a graph $G_i = \langle V, E_i \rangle$ with some set of edges E_i. This gives rise to a *profile* of graphs $\boldsymbol{G} = (G_1, \ldots, G_n)$, which we shall also write as $\boldsymbol{G} = \langle V, (E_1, \ldots, E_n) \rangle$. An *aggregator* is a function $F : \mathcal{G}^{\mathcal{N}} \to \mathcal{G}$ mapping any such profile into a single collective graph.

We require a few further pieces of *notation:* First, $E(x) := \{y \in V \mid (x, y) \in E\}$ is the set of successors of a vertex x in a set of edges E; and $E^{-1}(y) := \{x \in V \mid (x, y) \in E\}$ is the

Property	First-order Condition
Reflexivity	$\forall x.xEx$
Irreflexivity	$\neg\exists x.xEx$
Symmetry	$\forall xy.(xEy \rightarrow yEx)$
Antisymmetry	$\forall xy.(xEy \wedge yEx \rightarrow x = y)$
Transitivity	$\forall xyz.(xEy \wedge yEz \rightarrow xEz)$
Euclidean property	$\forall xyz.(xEy \wedge xEz \rightarrow yEz)$
Church-Rosser property	$\forall xy.[xEy \wedge xEz \rightarrow \exists w.(yEw \wedge zEw)]$
Seriality	$\forall x.\exists y.xEy$
Functionality	$\forall xyz.(xEy \wedge xEz \rightarrow y = z)$
Completeness	$\forall xy.[x \neq y \rightarrow (xEy \vee yEx)]$
Strong completeness	$\forall xy.(xEy \vee yEx)$
Connectedness	$\forall xyz.[xEy \wedge xEz \rightarrow (yEz \vee zEy)]$
Negative transitivity	$\forall xyz.[xEy \rightarrow (xEz \vee zEy)]$

Table 1: Common Properties of Directed Graphs.

set of predecessors of y. Second, given an edge e, we sometimes write $e \in G$ instead of $e \in E$ when $G = \langle V, E\rangle$. Third, xEy is a shorthand for $(x, y) \in E$. Fourth, $N_e^{\boldsymbol{G}} := \{i \in \mathcal{N} \mid e \in E_i\}$ is the set of individuals accepting edge e under profile $\boldsymbol{G}$.

A few fundamental *properties* of directed graphs (and, more generally speaking, of binary relations) are shown in Table 1. Recall that a *weak order* is a binary relation that is reflexive, transitive and complete, while a *linear order* is irreflexive, transitive and complete.

2.1 Properties of Graph Aggregators

We now introduce a number of axioms that define certain desirable properties of aggregators. The first such axiom is an independence condition that requires that the decision of whether or not a given edge e should be part of the collective graph should only depend on which of the individual graphs include e. This corresponds to the well-known independence axioms in preference aggregation (Arrow, 1963) and judgment aggregation (List and Puppe, 2009).

Definition 1 (IIE)**.** *F is independent of irrelevant edges if $N_e^{\boldsymbol{G}} = N_e^{\boldsymbol{G'}}$ implies $e \in F(\boldsymbol{G}) \Leftrightarrow e \in F(\boldsymbol{G'})$.*

That is, if exactly the same individuals accept e under profiles $\boldsymbol{G}$ and $\boldsymbol{G'}$, then e should be part of either both or none of the corresponding collective graphs. Note that above definition applies to *all* edges $e \in V^2$ and *all* pairs of profiles $\boldsymbol{G}, \boldsymbol{G'} \in \mathcal{G}^{\mathcal{N}}$. We shall leave this kind of universal quantification implicit also in later definitions.

While very much a standard axiom, we might be dissatisfied with IIE for not making reference to the fact that edges are defined in terms of vertices. Our next axiom is much more graph-specific and does not have a close analogue in preference or judgment aggregation. It requires that the decision of whether or not to collectively accept a given edge $e = (x, y)$ should only depend on which edges with the same source x are accepted by the individuals. Below we abuse notation and write $F(\boldsymbol{G})(x)$ for the set of successors of x in the set of edges in the collective graph $F(\boldsymbol{G})$ (and similarly $F(\boldsymbol{G})^{-1}(y)$ for the predecessors of y in $F(\boldsymbol{G})$).

Definition 2 (IIS)**.** *F is independent of irrelevant sources if $E_i(x) = E'_i(x)$ for all individuals $i \in \mathcal{N}$ implies $F(\boldsymbol{G})(x) = F(\boldsymbol{G'})(x)$.*

Similarly, the next axiom requires that collective acceptance of an edge $e = (x, y)$ should only depend on the pattern of individual acceptance for those edges with the same target y.

Definition 3 (IIT)**.** *F is independent of irrelevant targets if $E_i^{-1}(y) = {E'}_i^{-1}(y)$ for all individuals $i \in \mathcal{N}$ implies $F(\boldsymbol{G})^{-1}(y) = F(\boldsymbol{G'})^{-1}(y)$.*

Note that both IIS and IIT are strictly weaker than IIE. The precise relative strength of our independence axioms is illustrated by the following fact, which is easy to verify.

Fact 1. *An aggregator is IIE iff it is both IIS and IIT.*

The fundamental economic principle of *unanimity* requires that an edge should be accepted by the collective if all individuals accept it.

Definition 4 (Unanimity). *F is unanimous if $F(\boldsymbol{G}) = \langle V, E\rangle$ implies $E \supseteq E_1 \cap \cdots \cap E_n$.*

A requirement that, in some sense, is dual to unanimity is to ask that the collective graph should only include edges that are part of at least one of the individual graphs. In the context of ontology aggregation this axiom has been called *groundedness* (Porello and Endriss, 2011).

Definition 5 (Groundedness). *F is grounded if $F(\boldsymbol{G}) = \langle V, E\rangle$ implies $E \subseteq E_1 \cup \cdots \cup E_n$.*

The remaining axioms are all standard and closely modelled on their counterparts in judgment aggregation (List and Puppe, 2009).

Definition 6 (Anonymity). *F is anonymous if $F(G_1, \ldots, G_n) = F(G_{\pi(1)}, \ldots, G_{\pi(n)})$ for any permutation $\pi : \mathcal{N} \to \mathcal{N}$.*

Definition 7 (Neutrality). *F is neutral if $N_e^{\boldsymbol{G}} = N_{e'}^{\boldsymbol{G}}$ implies $e \in F(\boldsymbol{G}) \Leftrightarrow e' \in F(\boldsymbol{G})$.*

Definition 8 (Monotonicity). *F is monotonic if $e \in F(\boldsymbol{G})$ implies $e \in F(\boldsymbol{G'})$ whenever $\boldsymbol{G'}$ is obtained from $\boldsymbol{G}$ by having one additional individual accept the edge e.*

That is, anonymity and neutrality are basic symmetry requirements wrt. individuals and edges, respectively, while monotonicity requires that additional support for an edge should never reduce that edge's chances of being collectively accepted.

An extreme form of violating anonymity is to use an aggregator that is *dictatorial* in the sense that a single individual can determine the shape of the collective graph.

Definition 9 (Dictatorships). *F is dictatorial if there exists an individual $i^\star \in \mathcal{N}$ (the dictator) such that $e \in F(\boldsymbol{G}) \Leftrightarrow e \in G_{i^\star}$ for every edge $e \in V^2$.*

Aggregators that are not dictatorial are called *nondictatorial.*

Sometimes we shall only be interested in the properties of an aggregator as far as the *nonreflexive* edges $e = (x, y)$ with $x \neq y$ are concerned. Specifically, we call F *NR-neutral* if $N_{(x,y)}^{\boldsymbol{G}} = N_{(x',y')}^{\boldsymbol{G}}$ implies $(x, y) \in F(\boldsymbol{G}) \Leftrightarrow (x', y') \in F(\boldsymbol{G})$ for all $x \neq y$ and $x' \neq y'$; and we call F *NR-nondictatorial* if there exists no $i^\star \in \mathcal{N}$ such that $(x, y) \in F(\boldsymbol{G}) \Leftrightarrow (x, y) \in G_{i^\star}$ for all $x \neq y$. That is, NR-neutrality is slightly weaker than neutrality and NR-nondictatoriality is slightly stronger than nondictatoriality.

2.2 Aggregators

Next, we define several concrete aggregators. Under a *quota rule*, an edge will be included in the collective graph if the number of individuals accepting it meets a certain quota. If that quota is the same for every edge, then we have a *uniform* quota rule.

Definition 10 (Quota rules). *A quota rule is an aggregator F_q defined via a function $q : V^2 \to \{0, 1, \ldots, n{+}1\}$ by stipulating $F_q(\boldsymbol{G}) := \langle V, E\rangle$ with $E = \{e \in V^2 \mid |N_e^{\boldsymbol{G}}| \geqslant q(e)\}$. F_q is called uniform if q is a constant function.*

The class of uniform quota rules includes several interesting special cases:

- The (strict) *majority rule* accepts an edge if more than half of the individuals do. This is the uniform quota rule with $q = \lceil \frac{n+1}{2} \rceil$.
- The *union rule* is the aggregator that maps any given profile of graphs to their union: $\langle V, E_1 \cup \cdots \cup E_n \rangle$. This is the uniform quota rule with $q = 1$.
- The *intersection rule* is the aggregator that maps any given profile of graphs to their intersection: $\langle V, E_1 \cap \cdots \cap E_n \rangle$. This is the uniform quota rule with $q = n$.

We call the uniform quota rules with $q = 0$ and $q = n{+}1$ the *trivial* quota rules; $q = 0$ means that *all* edges will be included in the collective graph and $q = n{+}1$ means that *no* edge will be included. Quota rules have also been studied in judgment aggregation (Dietrich and List, 2007).

Another important class of aggregators, familiar from both judgment aggregation and belief merging, are the *distance-based aggregators* (Konieczny and Pino Pérez, 2002), which in our context amount to selecting a collective graph that satisfies certain properties and that minimises the distance to the individual graphs (for a suitable notion of distance and a suitable form of aggregating such distances). While of great practical importance, we shall not consider distance-based aggregators here, because they ensure that the collective graph meets the required properties "by design", i.e., the question of collective rationality does not arise for these rules. Distance-based rules also violate several attractive axioms (Lang et al., 2011) and are of high complexity (Endriss et al., 2010).

Inspired by *approval voting* (Brams and Fishburn, 2007), we now introduce a new class of aggregators specifically for graphs. Imagine we associate with each vertex an election in which all the possible successors of that vertex are the candidates (and in which there may be more than one winner). Each individual votes by stating which vertices they consider acceptable successors. We might then elect those vertices that receive the most support or that receive above average support. We might also give each voter a certain weight, which could be inversely proportional to the number of successors they propose, and so forth.

Definition 11 (Successor-approval rules)**.** *A successor-approval rule is an aggregator F_v defined via a function $v : (2^V)^{\mathcal{N}} \to 2^V$ by stipulating $F(\langle V, E_1, \ldots, E_n \rangle) := \langle V, E \rangle$ with $E = \{(x, y) \in V^2 \mid y \in v(E_1(x), \ldots, E_n(x))\}$.*

We call v the *choice function* associated with F_v. We shall only be interested in choice functions v that are *anonymous* and *neutral*, i.e., that satisfy $v(S_1, \ldots, S_n) = v(S_{\pi(1)}, \ldots, S_{\pi(n)})$ for any permutation $\pi : \mathcal{N} \to \mathcal{N}$ and for which $\{i \in \mathcal{N} \mid e \in S_i\} = \{i \in \mathcal{N} \mid e' \in S_i\}$ entails $e \in v(S_1, \ldots, S_n) \Leftrightarrow e' \in v(S_1, \ldots, S_n)$.

2.3 Characterisations

A simple adaptation of a result by Dietrich and List (2007) yields:

Fact 2. *An aggregator is a quota rule iff it is anonymous, IIE and monotonic.*

If we add the axiom of neutrality, then we obtain the class of uniform quota rules. If we furthermore impose unanimity and groundedness, then this excludes the trivial quota rules. Similarly, IIS characterises the class of successor-approval rules:

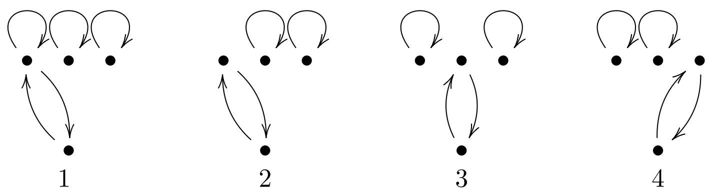
Fact 3. *An aggregator is a successor-approval rule (with an anonymous and neutral choice function) iff it is anonymous, neutral and IIS.*

3 Collective Rationality

We now analyse to what extent aggregators can ensure that a given property that is satisfied by each of the individual graphs is preserved when we move to the corresponding collective graph. This is known as *collective rationality.* For instance, in preference aggregation we may ask whether an aggregator can guarantee that the structure it will produce as output, when all the input structures are transitive and complete preference orders, will also be transitive and complete. Arrow's Theorem shows that the answer to this question is negative for any "reasonable" aggregator (Arrow, 1963). Much of this line of work has concentrated on properties that are natural to consider in the context of preference modelling. In our own previous work on binary aggregation, we have concentrated on properties that can be expressed in simple logical languages (Grandi and Endriss, 2010). Here, instead, we focus on fundamental properties of binary relations and graphs.

Definition 12 (Collective rationality)**.** *An aggregator F is collectively rational (CR) wrt. a property P if $F(\boldsymbol{G})$ satisfies P whenever each of the individual graphs in the profile $\boldsymbol{G}$ do.*

Example 1. *Suppose four individuals each provide a graph over the same set of vertices:*



If we apply the strict majority rule, we obtain a graph where the only edges are those connecting the upper three worlds with themselves. That is, this rule is not CR wrt. seriality, because each of the individual graphs is serial, while the collective graph computed is not. Symmetry, on the other hand, is preserved. There also is no violation of collective rationality wrt. reflexivity, because the individual graphs are not reflexive to begin with. A rule that does preserve seriality is the simple successor-approval rule that accepts an edge if it is (tied for being) most often accepted amongst those with the same source.

3.1 Basic Results

We begin with two very simple positive results, showing how a basic aggregation axiom can guarantee the preservation of a simple graph property:

Proposition 4. *Any unanimous aggregator is CR wrt. reflexivity.*

Proof. Immediate: If every individual graph includes all edges of the form (x, x), then unanimity ensures that the same is true for the collective graph. □

Proposition 5. *Any grounded aggregator is CR wrt. irreflexivity.*

Proof. Immediate: If no individual graph includes the edge (x, x), then groundedness guarantees the same for the collective graph. □

Symmetry is more demanding a property and unanimity alone does not suffice to preserve it. However, if we restrict attention to uniform quota rules, we obtain the following result:

Proposition 6. *Any uniform quota rule is CR wrt. symmetry.*

Proof. Immediate: If each individual respects symmetry, then the number of individual graphs including edge (x, y) will always equal the number of individual graphs including (y, x). Hence, either both or neither will meet the uniformly imposed quota. □

Note that uniformity is a necessary condition for Proposition 6 to hold. Transitivity is yet again more demanding a property:

Proposition 7. *The intersection rule is CR wrt. transitivity. It is the only nontrivial uniform quota rule with that property for* $|V| \geqslant 3$.

Proof. First, it is easy to verify that the intersection rule preserves transitivity. Now consider any nontrivial uniform quota rule F_q with a quota $q < n$. Take a profile in which $q - 1$ individuals accept (x, y), (y, z) and (x, z); one individual accepts only (x, y); and one individual accepts only (y, z). This profile is transitive (as far as the edges under consideration here are concerned). But when we aggregate using F_q, then we obtain a graph that includes the edges (x, y) and (y, z), but not (x, z). Hence, transitivity is not preserved. □

The constant rules corresponding to the trivial quota rules with $q = 0$ and $q = n+1$ vacuously ensure collective rationality wrt. transitivity. Another demanding property is seriality:

Proposition 8. *The union rule is CR wrt. seriality. It is the only nontrivial uniform quota rule with that property for* $|V| \geqslant n$.

Proof. Clearly, the union rule (with $q = 1$) will preserve seriality. To see that no uniform quota rule with $1 < q \leqslant n$ does, it suffices to consider a scenario where each of the edges emanating from a particular source x is accepted by (at most) one individual. Note that this construction requires $|V| \geqslant n$. Otherwise, there always is an outgoing edge accepted by more than one individual (when each individual respects seriality), and therefore also some quotas $q > 1$ will work. □

Amongst the trivial uniform quota rules only the one with $q = 0$ ensures seriality. If we move away from quota rules (satisfying IIE) and are content with using successor-approval rules (only satisfying IIS), then we have a wider choice of aggregators available that will preserve seriality (e.g., the simple successor-approval rule of Example 1).

Above we have seen that certain properties will be preserved by certain quota rules. However, if we want to preserve several such properties, those possibility results quickly turn into impossibilities. Let us begin with an immediate corollary of our earlier results:

Corollary 9. *If* $|V| \geqslant n$, *then no nontrivial uniform quota rule is CR wrt. both transitivity and seriality.*

Proof. Immediate from Proposition 7 and Proposition 8 and the fact that union and intersection differ for $n > 1$. □

Rather surprisingly, in some cases we obtain an impossibility already when only collective rationality wrt. a *single* property is required:

Proposition 10. *If* $|V| \geqslant 3$, *then no nontrivial uniform quota rule is CR wrt. connectedness.*

Proof. First, the intersection rule (with quota $q = n$) does not preserve connectedness. To see this, consider a scenario where all individuals accept (x, y) and (x, z), half of them accept (y, z), and the other half (z, y). Second, for any uniform quota rule with $0 < q < n$, construct a counterexample as follows: Suppose a group of q individuals accept (x, y), a different group of q individuals accept (x, z), and their intersection accept (y, z), while nobody accepts (z, y). Then (x, y) and (x, z) are part of the collective graph, but neither (y, z) or (z, y) are. This violates connectedness, even though the individual graphs satisfy it. □

Note that both of the trivial uniform quota rules ensure connectedness (because both the complete and the empty graph are connected). If we swap connectedness for completeness, then we obtain the following characterisation:

Proposition 11. *If $|V| \geqslant 2$, then a uniform quota rule F_q is CR wrt. completeness (or strong completeness) iff $q \leqslant \lfloor \frac{n+1}{2} \rfloor$.*

Proof. By the pigeon hole principle, if all individual graphs are complete, then one of (x, y) and (y, x) will always have at least $\lfloor \frac{n+1}{2} \rfloor$ individuals accepting it. Hence, if (and only if) the quota is at most $\lfloor \frac{n+1}{2} \rfloor$ we can ensure that that edge will be collectively accepted. □

While most of our examples so far have been restricted to quota rules, they already give some insight into the close connections between collective rationality and standard axiomatic requirements. In the sequel, we shall explore this connection in much more depth.

3.2 Impossibility Theorems

In view of Fact 2 and the remarks following it, we can reformulate Proposition 10 as saying that there exists no anonymous, neutral, unanimous, grounded, IIE and monotonic aggregator that is CR wrt. connectedness. This closely resembles classical impossibility theorems in social choice theory. For instance, Arrow's Theorem in its form for linear orders (i.e., irreflexive, transitive, and complete preference orders) can be stated as saying that there exists no nondictatorial, unanimous, and IIE aggregator that is CR wrt. irreflexivity, transitivity, and completeness. We shall soon prove the following variant of Arrow's Theorem:[2]

Theorem 12. *If $|V| \geqslant 3$, then there exists no NR-nondictatorial, unanimous, grounded and IIE aggregator that is CR wrt. both transitivity and completeness.*

For now, we want to see whether Arrow's impossibility persists when we move away from properties typically associated with preferences. The central axiom the impossibility feeds on is IIE. Observe that an aggregator F satisfies IIE iff for each edge $e \in V^2$ there exists a set of *winning coalitions* $\mathcal{W}_e \subseteq 2^{\mathcal{N}}$ such that $e \in F(\boldsymbol{G}) \Leftrightarrow N_e^{\boldsymbol{G}} \in \mathcal{W}_e$. Imposing additional axioms on F corresponds to restrictions on the associated family of winning coalitions, e.g.:

- If F is unanimous, then $\mathcal{N} \in \mathcal{W}_e$ for any edge e.
- If F is grounded, then $\emptyset \notin \mathcal{W}_e$ for any edge e.
- If F is (NR-)neutral, then $\mathcal{W}_e = \mathcal{W}_{e'}$ for any two (nonreflexive) edges e and e'.

Recall that neutrality does not feature in Arrow's Theorem. As we shall see next, the reason is that the same restriction on winning coalitions is already enforced by collective rationality wrt. transitivity (at least for nonreflexive edges). This is a surprising and interesting link between a specific collective rationality requirement and a specific axiom. This link is related to the so-called *Contagion Lemma* (Sen, 1986), but we have not seen it noted in the literature in this form before. The same kind of result can also be obtained for other graph properties with a similar structure; besides transitivity, we state it here for the Euclidean property.

Lemma 13. *If $|V| \geqslant 3$, then any unanimous and IIE aggregator that is CR wrt. transitivity or the Euclidean property must be NR-neutral.*

Proof. Let F be an aggregator that is unanimous and IIE, and let $\{\mathcal{W}_e\}_{e \in V^2}$ be the associated family of winning coalitions. We need to show that there exists a unique $\mathcal{W} \subseteq 2^{\mathcal{N}}$

[2] Theorem 12 implies both the standard variant of Arrow's Theorem for linear orders and its standard variant for weak orders. (1) For linear orders: First, by Proposition 5 we can add irreflexivity to the CR requirements without changing the logical strength of the theorem. Second, groundedness can be dropped as it follows from unanimity together with CR wrt. completeness. Third, on irreflexive profiles NR-nondictatoriality and nondictatoriality coincide. (2) For weak orders: First, by Proposition 4 we can add reflexivity to the CR requirements. Second, the Pareto condition (i.e., unanimity wrt. the strict part of the preference relation) implies both unanimity (when used with IIE) and groundedness (when used with CR wrt. completeness). Third, on reflexive profiles NR-nondictatoriality and nondictatoriality coincide.

such that $\mathcal{W} = \mathcal{W}_e$ for any nonreflexive edge e. Note that the $\mathcal{W}_e$ are not empty (due to unanimity). Consider any three vertices x, y, z and any coalition $C \in \mathcal{W}_{(x,y)}$. We will employ collective rationality to show that C must also be a winning coalition for each of the other five edges between these three vertices. A simple inductive argument then suffices to show that C will in fact have to be a winning coalition for *all* nonreflexive edges.

Now suppose F is CR wrt. transitivity. Let us first see how to prove that $C \in \mathcal{W}_{(z,x)}$: Consider a scenario in which (x, y) and (z, x) are accepted by the individuals in C and only those (i.e., by definition of C, (x, y) is collectively accepted) and in which (y, z) is accepted by all individuals (i.e., by unanimity, (y, z) is also collectively accepted). Then, by collective transitivity, (z, x) must be collectively accepted. Hence, C must be a winning coalition for (z, x), i.e., $C \in \mathcal{W}_{(z,x)}$. We can use a similar argument for the other edges: e.g., to show $C \in \mathcal{W}_{(z,y)}$ consider the case with C accepting all of (z, x), (x, y) and (z, y); to show $C \in \mathcal{W}_{(y,x)}$ consider the case with everyone accepting (y, z) and C accepting (z, x) and (y, x); and so forth.

The proof in case transitivity is replaced by the Euclidean property is similar. We omit the details in the interest of space. □

Note that Lemma 13 does not hold for $|V| = 2$: the aggregator that accepts (x, y) whenever agent 1 does and that accepts (y, x) whenever agent 2 does is a counterexample.

Also note that full neutrality does not follow from the conditions of Lemma 13. The reason is that, while C being a winning coalition for (x, y) entails C also being a winning coalition for (x, x), the converse is not true. For example, the aggregator that accepts nonreflexive edges only when all individuals do, but that always accepts all reflexive edges (thereby violating neutrality), is unanimous, IIE, and CR wrt. transitivity.

We now prove a result similar to Arrow's Theorem, but replacing completeness by seriality. We do this by proving that the set of winning coalitions corresponding to any aggregator that meets the conditions stated in the theorem is an *ultrafilter* (Davey and Priestley, 2002).

Definition 13 (Ultrafilters.)**.** *An ultrafilter $\mathcal{W}$ on a set $\mathcal{N}$ is a collection of subsets of $\mathcal{N}$ that satisfies the following three conditions:*

(i) $\emptyset \notin \mathcal{W}$;
(ii) $C_1, C_2 \in \mathcal{W}$ *implies* $C_1 \cap C_2 \in \mathcal{W}$ *(i.e., $\mathcal{W}$ is closed under intersections); and*
(iii) C *or* $\mathcal{N} \setminus C$ *is in* $\mathcal{W}$ *for any* $C \subseteq \mathcal{N}$ *(i.e., $\mathcal{W}$ is maximal).*

We are now ready to state and prove our result:

Theorem 14. *If $|V| \geqslant 3$, then there exists no NR-nondictatorial, unanimous, grounded, and IIE aggregator that is CR wrt. both transitivity and seriality.*

Proof. Let F be a unanimous, grounded and IIE aggregator that is CR wrt. transitivity and seriality. By Lemma 13, F is NR-neutral, i.e., there is set of winning coalitions $\mathcal{W} \subseteq 2^{\mathcal{N}}$ with $e \in F(\boldsymbol{G}) \Leftrightarrow N_e^{\boldsymbol{G}} \in \mathcal{W}$ for any nonreflexive edge e. We shall prove that $\mathcal{W}$ is an ultrafilter. Condition (i) holds, because F is grounded. Condition (ii) follows from collective rationality wrt. transitivity: Suppose $C_1, C_2 \in \mathcal{W}$. Consider a scenario where coalition C_1 accepts (x, y) and C_2 accepts (y, z). Then, by transitivity, *at least* coalition $C_1 \cap C_2$ must accept (x, z). Suppose it is exactly the individuals in $C_1 \cap C_2$ who do. As C_1 and C_2 are winning coalitions, (x, y) and (y, z) are part of the collective graph. To achieve collective rationality wrt. transitivity, we must also have (x, z) be part of the collective graph, and thus we must have $C_1 \cap C_2 \in \mathcal{W}$. Condition (iii), finally, follows from collective rationality wrt. seriality: Take an arbitrary coalition $C \in \mathcal{W}$. Consider a scenario where exactly the individuals in C accept (x, y), exactly those in $\mathcal{N} \setminus C$ accept (x, z), and no individual accepts any of the other edges emanating from x. Due to groundedness, of all the edges emanating from x, only (x, y) and (x, z) can possibly be part of the collective graph. Due to collective rationality wrt. seriality at least one of them has to be, i.e., $C \in \mathcal{W}$ or $\mathcal{N} \setminus C \in \mathcal{W}$.

Recall that $\mathcal{N}$ is required to be finite. An ultrafilter $\mathcal{W}$ on a set $\mathcal{N}$ is called *principal* if it is of the form $\mathcal{W} = \{C \in 2^{\mathcal{N}} \mid i^\star \in C\}$ for some fixed $i^\star \in \mathcal{N}$. In our setting, principality of $\mathcal{W}$ corresponds to F being dictatorial (with dictator $i^\star$) on nonreflexive edges. Now, it is a well-known fact that any ultrafilter on a finite set must be principal (Davey and Priestley, 2002), which shows that F cannot be NR-nondictatorial. □

We can obtain a proof of Arrow's Theorem, in our rendering as Theorem 12, using the very same approach. Above, we used seriality only to establish condition (*iii*). We can use completeness, featuring in Theorem 12, instead: simply consider a scenario in which all individuals in C accept (x, y) and all those in $\mathcal{N} \setminus C$ accept (y, x). Then one of C and $\mathcal{N} \setminus C$ must be a winning coalition to ensure completeness for the collective graph. This observation completes the proof of Arrow's Theorem (Theorem 12).

To demonstrate the versatility of our approach, let us state one more impossibility:

Theorem 15. *If $|V| \geqslant 3$, then there exists no NR-nondictatorial, unanimous, grounded, and IIE aggregator that is CR wrt. both the Euclidean property and seriality.*

Proof. In our proof of Theorem 14, we used collective rationality wrt. transitivity twice: to invoke Lemma 13 and to establish ultrafilter condition (*ii*). Lemma 13 still applies when we use the Euclidean property instead of transitivity. So we only need to prove condition (*ii*): Suppose only agents in C_1 accept (x, y), only those in C_2 accept (x, z), and only those in $C_1 \cap C_2$ accept (y, z). That is, all individual graphs satisfy the Euclidean property (wrt. x, y and z). If both C_1 and C_2 are winning coalitions, then the collective graph will include (x, y) and (x, z). To satisfy the Euclidean property, it will also have to include (y, z). Hence, $C_1 \cap C_2$ must also be a winning coalition. □

How interesting Theorems 14 and 15 are is open to debate. Certainly, neither of them has the immediate intuitive appeal of Arrow's Theorem, which speaks about a class of graphs that can be interpreted as preference orders. On the other hand, these results indicate a generic technique for proving impossibility results in the style of Arrow's Theorem by explicitly linking (a) specific properties wrt. which we want to impose collective rationality and (b) specific conditions on ultrafilters. We obtain the following general picture:

(1) The *condition of closure-under-intersections* of an ultrafilter ($C_1, C_2 \in \mathcal{W} \Rightarrow C_1 \cap C_2 \in \mathcal{W}$) is derivable from collective rationality wrt. to any one of the following graph properties: *transitivity*, the *Euclidean property*, and the *Church-Rosser property*.[3] What these properties have in common is that they force the acceptance of one edge (or two, in the case of Church-Rosser) given the acceptance of two other edges. Any other graph property with this feature can be applied to the same effect.

(2) The *condition of maximality* of an ultrafilter ($C \in \mathcal{W}$ or $\mathcal{N} \setminus C \in \mathcal{W}$) is derivable from collective rationality wrt. to any one of the following graph properties: *completeness*, *strong completeness*, *connectedness*, *negative transitivity*, and *seriality*. What they have in common is that they force the acceptance of at least one out of a set of several (usually two) edges, possibly given the acceptance of some other edges (for connectedness and negative transitivity). Any other graph property with this feature can be applied to the same effect.

(3) Collective rationality wrt. graph properties that either do not create dependencies between edges (such as reflexivity or irreflexivity) or that do not force the acceptance of

[3] Church-Rosser requires 4 (rather than just 3) vertices to be applicable.

at least one edge (such as symmetry, antisymmetry, or functionality) cannot contribute to establishing the ultrafilter conditions.

The ultrafilter method itself becomes applicable once we assume IIE (needed to make winning coalitions applicable in the first place), neutrality (needed to show that all edges have the same winning coalitions), and unanimity (needed to show that the collection of winning coalitions is not empty). Groundedness is needed for the first ultrafilter condition. That is, we obtain an Arrovian impossibility for graph aggregation as soon as we accept these four axioms and postulate collective rationality wrt. one property from the first group above and one property from the second property above. As we have seen in Lemma 13, rather than accepting neutrality from the outset, we can also derive it as a consequence of collective rationality wrt. certain graph properties.

4 Conclusions, Related Work and Future Directions

We have argued that graph aggregation is an important problem with several potential applications and we have introduced a simple formal framework to study this problem. We have defined quota rules and successor-approval rules as interesting aggregators and we have stated several natural axiomatic requirements. Finally, we have argued that collective rationality is of central importance in the study of (not only!) this type of aggregation problem. Our main technical contribution has been a refinement of the ultrafilter method, allowing us to approach the proof of Arrovian impossibilities in a highly modular manner, clearly relating axiomatic properties, rationality properties, and ultrafilter properties.

Our approach is also helpful in interpreting a recent result by Pini et al. (2009), who prove a variant of Arrow's Theorem for preorders, i.e., for preferences that need not be complete. To be able to prove their result, these authors require the collective preference order to have one element that is weakly preferred (or dispreferred) to all other elements. This may be interpreted as a (very weak) form of completeness. Indeed, that such a condition would be needed is exactly what we would expect in view of our analysis above (without it, we cannot obtain the third ultrafilter condition) and it is not hard to see how to adapt our proof of Theorem 12 to provide a new simple proof of the result of Pini et al. (2009).

In related work on belief merging, Maynard-Zhang and Lehmann (2003) suggest an approach to circumvent Arrow's Theorem by (a) replacing completeness by negative transitivity (which they call "modularity") and (b) weakening the independence axiom. In the discussion of their result, these authors stress the significance of both of these changes. However, our analysis easily shows that replacing completeness by negative transitivity alone has no effect on Arrow's impossibility: the maximality condition of an ultrafilter can easily be derived using collective rationality wrt. negative transitivity (just consider a scenario where everyone in C accepts (x, z) and everyone else accepts (z, y)). Hence, the crucial source for the possibility result of Maynard-Zhang and Lehmann must be their modification of the independence axiom (and, indeed, this modification is rather substantial as it allows for independence to be violated whenever not doing so would lead to a "conflict").

Other related work includes our own work on collective rationality in binary aggregation, where we link axiomatic properties and structural properties of the integrity constraints used to define rationality assumptions in a propositional language (Grandi and Endriss, 2010), and so-called *agenda characterisation theorems* in judgment aggregation, linking axiomatic properties and collective rationality wrt. logical consistency (List and Puppe, 2009).

An interesting direction for future work that we have begun to explore is to study collective rationality wrt. the truth of a formula in *modal logic* evaluated over a directed graph. A basic result here shows that an aggregator F is grounded iff F is CR wrt. any modal formula not involving a $\Diamond$-operator (or a $\Box$-operator within the scope of a negation).

References

S. Airiau, U. Endriss, U. Grandi, D. Porello, and J. Uckelman. Aggregating dependency graphs into voting agendas in multi-issue elections. In *Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, 2011.

K. J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 2nd edition, 1963.

R. Bradley, F. Dietrich, and C. List. Aggregating causal judgments. Working paper, London School of Economics, 2011.

S. J. Brams and P. C. Fishburn. *Approval Voting*. Springer, 2nd edition, 2007.

S. Coste-Marquis, C. Devred, S. Konieczny, M.-C. Lagasquie-Schiex, and P. Marquis. On the merging of Dung's argumentation systems. *Artificial Intelligence*, 171(10–15):730–753, 2007.

B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2nd edition, 2002.

F. Dietrich and C. List. Judgment aggregation by quota rules: Majority voting generalized. *Journal of Theoretical Politics*, 19(4):391–424, 2007.

E. Dokow and R. Holzman. Aggregation of binary evaluations. *Journal of Economic Theory*, 145 (2):495–511, 2010.

P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. 10th International World Wide Web Conference (WWW-2001)*, 2001.

U. Endriss, U. Grandi, and D. Porello. Complexity of winner determination and strategic manipulation in judgment aggregation. In *Proc. 3rd International Workshop on Computational Social Choice (COMSOC-2010)*, 2010.

U. Grandi and U. Endriss. Lifting rationality assumptions in binary aggregation. In *Proc. 24th AAAI Conference on Artificial Intelligence (AAAI-2010)*, 2010.

F. Herzberg and D. Eckert. The model-theoretic approach to aggregation: Impossibility results for finite and infinite electorates. *Mathematical Social Sciences*, 2011. In press.

M. O. Jackson. *Social and Economic Networks*. Princeton University Press, 2008.

A. Kirman and D. Sondermann. Arrow's theorem, many agents, and invisible dictators. *Journal of Economic Theory*, 5(2):267–277, 1972.

S. Konieczny and R. Pino Pérez. Merging information under constraints: A logical framework. *Journal of Logic and Computation*, 12(5):773–808, 2002.

J. Lang, G. Pigozzi, M. Slavkovik, and L. van der Torre. Judgment aggregation rules based on minimization. In *Proc. 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-XIII)*, 2011.

C. List and C. Puppe. Judgment aggregation: A survey. In *Handbook of Rational and Social Choice*. Oxford University Press, 2009.

P. Maynard-Zhang and D. J. Lehmann. Representing and aggregating conflicting beliefs. *Journal of Artificial Intelligence Research (JAIR)*, 19:155–203, 2003.

M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Aggregating partially ordered preferences. *Journal of Logic and Computation*, 19(3):475–502, 2009.

D. Porello and U. Endriss. Ontology merging as social choice. In *Proc. 12th International Workshop on Computational Logic in Multiagent Systems (CLIMA-2011)*. Springer, 2011.

A. K. Sen. Social choice theory. In *Handbook of Mathematical Economics*, volume 3. North-Holland, 1986.

Ulle Endriss and Umberto Grandi
Institute for Logic, Language and Computation
University of Amsterdam
Postbus 94242, 1090 GE Amsterdam, The Netherlands
Email: `ulle.endriss@uva.nl`, `umberto.uni@gmail.com`

Manipulation Under Voting Rule Uncertainty[1]

Edith Elkind and Gábor Erdélyi

Abstract

An important research topic in the field of computational social choice is the complexity of various forms of dishonest behavior, such as manipulation, control, and bribery. While much of the work on this topic assumes that the cheating party has full information about the election, recently there have been a number of attempts to gauge the complexity of non-truthful behavior under uncertainty about the voters' preferences. In this paper, we analyze the complexity of (coalitional) manipulation for the setting where there is uncertainty about the voting rule: the manipulator(s) know that the election will be conducted using a voting rule from a given list, and need to select their votes so as to succeed no matter which voting rule will eventually be chosen. We identify a large class of voting rules such that arbitrary combinations of rules from this class are easy to manipulate; in particular, we show that this is the case for single-voter manipulation and essentially all easy-to-manipulate voting rules, and for coalitional manipulation and k-approval. While a combination of a hard-to-manipulate rule with an easy-to-manipulate one is usually hard to manipulate—we prove this in the context of coalitional manipulation for several combinations of prominent voting rules—we also provide counterexamples showing that this is not always the case.

1 Introduction

Voting is an established framework for making collective decisions, and as such has applications in settings that range from political elections to faculty hiring decisions, selecting the winners of singing competitions, and the design of multiagent systems. In some of these settings, the number of candidates and/or voters can be large, yet the decision needs to be made quickly. Whenever this is the case, the algorithmic complexity of, on the one hand, winner determination and, on the other hand, various forms of dishonest behavior in elections, plays an important role in the selection of a voting rule: we want the former to be as low as possible, while keeping the latter as high as possible.

Traditionally, the complexity of voting rules is studied under the full information assumption: for instance, in the single-voter manipulation problem, which is perhaps one of the most fundamental problems in the complexity-theoretic analysis of voting rules, it is assumed that the manipulator knows the set of candidates, the number and the true preferences of all honest voters, and, crucially, the voting rule. However, it is widely recognized that this assumption is not always realistic, and recently a number of papers tried to analyze the complexity of cheating in elections and/or determining the likely election winners under various forms of uncertainty about the election (see Section 1.1 for an overview).

In this paper, we study the complexity of manipulation (both by a single voter and by a coalition of voters) in settings where there is uncertainty about the voting rule itself. That is, we assume that the manipulator(s) know that the voting rule belongs to a certain (finite or infinite) family of rules $\widehat{\mathcal{F}}$, and they want to select their votes so as to ensure that their preferred candidate wins, no matter which of the rules in $\widehat{\mathcal{F}}$ is chosen.

Admittedly, in political elections the voting rule to be used is typically known before the votes are cast, and the manipulator would be well advised to fully understand the voting rule before modifying her vote. However, in other applications of voting this is not always the case. For instance, it is not unusual for a university department to ask graduate students to provide a ranking of faculty

[1]This research was supported by National Research Foundation (Singapore) under grant 2009-08. Work done in part while the second author was affiliated with Nanyang Technological University, Singapore. A previous version of this paper was presented at AAMAS'12.

candidates; however, the graduate students are not told how the hiring committee makes its decision (anecdotally, a wide variety of voting rules can be used for this purpose). Another example is provided by conference reviewing: at some point in the decision-making process, the program committee members may be asked to rank the papers whose fate has not been decided yet; the PC chair will then aggregate the rankings in a way that has not been announced to the PC members (and may, in fact, be unknown to the PC chair when she initiates the process). In some of these settings, the voters may believe that the voting rule will be chosen from a specific family of rules: for instance, the voters may know that the rule to be used is a scoring rule, or, more narrowly, a k-Approval rule (with the value of k unknown), or a Condorcet-consistent rule (see Section 2 for definitions); the situation where the voters know the voting correspondence, but not the tie-breaking rule is also captured by this description. They may then want to select their votes so that their favorite candidate wins the election *no matter which of the voting rules in this family is chosen*.

We study the complexity of this problem for several families of voting rules. We limit ourselves to the setting of voting manipulation (either by a single voter or by a coalition of voters), though one can ask the same question in the context of election control or bribery (see, e.g., [13] for the definitions and a survey of recent results for these problems). We mostly focus on families that consist of a small number (usually, two) prominent voting rules, such as Plurality, k-Approval, Borda, Copeland, Maximin and STV. Our goal is not to classify all such combinations or rules: rather, we try to illustrate the general techniques that can be used for the analysis of such settings.

One would expect a combination of easy-to-manipulate rules to be easy to manipulate, and a combination of several hard-to-manipulate rules or an easy-to-manipulate one with a hard-to-manipulate one to be hard to manipulate. Our results for classic voting rules mostly confirm this intuition, with the exception of settings where we combine a hard-to-manipulate rule with one that is very indecisive. However, we show that these results are not universal: we provide an example of two hard-to-manipulate rules whose combination is easy to manipulate, as well as an example of two easy-to-manipulate rules whose combination is hard to manipulate. While the rules used in these constructions are fairly artificial, they nevertheless illustrate interesting aspects of our problem.

1.1 Related Work

Our works fits into the stream of research on winner determination and voting manipulation under uncertainty. In the context of winner determination, perhaps the most prominent problem in this category is the possible/necessary winner problem [16], where the voting rule is public information, but, for each voter, only a partial order over the candidates in known; the goal is to determine if a candidate wins the election for *some* way (the *possible winner*) or for *every* way (the *necessary winner*) of completing the voters' preferences; a probabilistic variant of this problem has also been considered [1]. Our problem is more similar in flavor to the necessary winner problem, as the manipulator has to succeed for *all* voting rules in the family.

Uncertainty about the voting rule has been recently investigated by Baumeister et al. [5], who also consider the situation where the voting rule will be chosen from a fixed set. In contrast to our work, they assume that all voters' preferences are known, and ask if there is a voting rule that makes a certain candidate a winner with respect to these preferences; thus, in their work the manipulating party is the election authority rather than one of the voters.

Our problem is, in a sense, dual to the one considered by Conitzer et al. [7]: in their model the voting rule is known, but the preferences of some of the honest voters are (partially) unknown; they ask if the manipulator can cast a vote that improves the outcome (from his perspective) for *every* realization of the honest voters' preferences; thus, just like us, they assume an adversarial environment.

There has also been some work on settings where the *effects of the manipulator's actions* are uncertain. This is the case, for instance, for the model of safe strategic voting [18], where one voter announces a manipulative vote, and one or more voters with the same true preferences may follow suit; the original manipulator does not know how many followers he will have and needs to choose the vote so as to improve the outcome for *some* number of followers, while ensuring that the outcome does not get worse for *any* number of followers. Another example is cloning [9], where the cheating party clones one or more candidates; the voters are assumed to rank the clones of a given candidate consecutively, but the exact order of the clones in voters' preferences is unknown. Our work is most similar to the variant of this problem known as 1-CLONING, where the cheating party has to succeed *no matter how the voters order the clones*.

Finally, we remark that the idea of combining two or more voting rules has been considered in early work on computational social choice [10, 14]; however, in both of these papers, voting rules are combined in a way that is very different from our work.

2 Preliminaries

Given a finite set S, we denote by $\mathcal{L}(S)$ the space of all linear orders over S. An *election* is a triple $E = (C, V, \mathcal{R})$, where $C = \{c_1, \dots, c_m\}$ is the set of *candidates*, V is the set of *voters*, $|V| = n$, and $\mathcal{R} = (R_1, \dots, R_n)$ is the *preference profile*, i.e., a collection of linear orders over C. The order R_i is called the *preference order*, or *vote*, of voter i; we will also denote R_i by $\succ_i$. When $a \succ_i b$ for some $a, b \in C$, we say that voter i *prefers* a to b. A candidate a is said to be the *top-ranked* candidate of voter i, or *receive a first-place vote* from i, if $a \succ_i b$ for all $b \in C \setminus \{i\}$.

A *voting correspondence* $\mathcal{F}$ is a mapping that, given an election $E = (C, V, \mathcal{R})$ outputs a non-empty subset $S \subseteq C$; we write $S = \mathcal{F}(E)$. The elements of the set S are called the *winners* of the election E under $\mathcal{F}$. If $|\mathcal{F}(E)| = 1$ for any election E, the mapping $\mathcal{F}$ is called a *voting rule*; whenever this is the case, we abuse notation and write $\mathcal{F}(\mathcal{R}) = c$ instead of $\mathcal{F}(\mathcal{R}) = \{c\}$. We will sometimes abuse terminology and refer to voting correspondences as voting rules.

A voting correspondence $\mathcal{F}$ is said to be *neutral* if renaming the candidates does not alter the set of winners: that is, for any election $E = (C, V, \mathcal{R})$ and any permutation π of the set C, the election E' obtained by replacing each candidate c in $\mathcal{R}$ by $\pi(c)$ satisfies $\mathcal{F}(E') = \{\pi(c) \mid c \in \mathcal{F}(E)\}$. $\mathcal{F}$ is said to be *monotone* if promoting a winning candidate does not make him lose the election: if $c \in \mathcal{F}(E)$, then $c \in \mathcal{F}(E')$, where E' is obtained from E by swapping c with the candidate ranked just above c in some vote (this notion of monotonicity is sometimes referred to as *weak* monotonicity).

Voting rules We will now describe the voting rules (correspondences) considered in this paper. For all rules that assign scores to candidates (i.e., scoring rules, Copeland, and Maximin), the winners are the candidates with the highest scores.

Scoring rules Any vector $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathbb{R}^m$ such that $\alpha_1 \geq \dots \geq \alpha_m$ defines a *scoring rule* $\mathcal{F}_\alpha$ over a set of candidates of size m: a candidate receives α_j points from each voter who ranks him in the j-th position, and the score of a candidate is the total number of points he receives from all voters. The vector α is called a *scoring vector*. We assume without loss of generality that the entries of α are nonnegative integers given in binary. As we require voting rules to be defined for any number of candidates, we will consider *families* of scoring rules: one for every possible number of candidates. We denote such families by $\{\mathcal{F}_{\alpha^m}\}_{m=1,\dots}$, where $\alpha^m = (\alpha_1^m, \dots, \alpha_m^m)$ is the scoring vector of length m. Two well-known examples of such families are Borda, given by $\alpha^m = (m-1, \dots, 1, 0)$ for all $m > 1$, and k-Approval, given by $\alpha_i^m = 1$ if $i \leq k$, $\alpha_i^m = 0$ if $i > k$. The 1-Approval rule is also known as Plurality.

Condorcet We say that a candidate a wins a *pairwise election* against b if more than half of the voters prefer a to b; if exactly half of the voters prefer a to b, then a is said to *tie* his pairwise election against b. A candidate is said to be a *Condorcet winner* if he wins pairwise elections against all other candidates. The Condorcet rule outputs the Condorcet winner if it exists; otherwise, it outputs the set of all candidates (recall that a voting correspondence should always output a non-empty set of winners).

Copeland Given a rational value $\alpha \in [0, 1]$, under the Copeland$^\alpha$ rule each candidate gets 1 point for each pairwise election he wins and α points for each pairwise election he ties.

Maximin The Maximin score of a candidate $c \in C$ is equal to the number of votes he gets in his worst pairwise election, i.e., $\min_{d \in C \setminus \{c\}} |\{i \mid c \succ_i d\}|$.

STV Under the STV rule, the election proceeds in rounds. During each round, the candidate with the lowest Plurality score is eliminated, and the candidates' Plurality scores are recomputed. The winner is the candidate that survives till the end. If several candidates have the lowest Plurality score (we will refer to this situation as an *intermediate tie*), we assume that the candidate to be eliminated is chosen according to the lexicographic order over the candidates: if S is the set of candidates that have the lowest Plurality score in some round, we eliminate the candidate c_j such that $j \geq i$ for all $c_i \in S$. We remark that STV, as defined here, always has a single winner; however, because of the lexicographic tie-breaking rule it is not neutral.

3 Problem Statement

We assume that we are given a collection $\widehat{\mathcal{F}} = \{\mathcal{F}_i\}_{i \in I}$ of voting correspondences. The set $\widehat{\mathcal{F}}$ can be finite of infinite; for instance, $\widehat{\mathcal{F}}$ can be the set of all (families of) scoring rules, in which case it is infinite. When $\widehat{\mathcal{F}}$ is infinite, we assume that it admits a succinct description; if $\widehat{\mathcal{F}}$ is finite, it is assumed to be listed explicitly.

We consider the complexity of (coalitional) manipulation in elections when the manipulator does not know which of the voting rules in $\widehat{\mathcal{F}}$ will be selected. We state our definitions in the *unique winner* model, i.e., we assume that the manipulator's goal is to make its preferred candidate the unique winner with respect to each of the voting correspondences in $\widehat{\mathcal{F}}$; however, most of our results remain true in the *co-winner* model, where the manipulator would like to ensure that its preferred candidate is one of the winners under each of the voting correspondences in $\widehat{\mathcal{F}}$.

Name: $\widehat{\mathcal{F}}$-MANIPULATION BY SINGLE VOTER (SM)

Input: An election (C, V) with $|C| = m$, $|V| = n - 1$, a preference profile $\mathcal{R} = (R_1, \ldots, R_{n-1})$, and a candidate $p \in C$.

Question: Is there a vote $L \in \mathcal{L}(C)$ such that p is the unique winner in $(\mathcal{R}, L)$ with respect to each of the voting correspondences in $\widehat{\mathcal{F}}$?

Voters $1, \ldots, n-1$ are referred to as the *honest* voters, and the last voter (the one who submits vote L and wants p to win) is referred to as the *manipulator*.

Name: $\widehat{\mathcal{F}}$-COALITIONAL MANIPULATION (CM)

Input: An election (C, V) with $|C| = m$, $|V| = h$, a set M, $|M| = s = n - h$, a preference profile $\mathcal{R} = (R_1, \ldots, R_h)$, and a candidate $p \in C$.

Question: Is there a profile $\mathcal{L} = (L_1, \ldots, L_s) \in \mathcal{L}^s(C)$ such that p is the unique winner in $(\mathcal{R}, \mathcal{L})$ with respect to each of the voting correspondences in $\widehat{\mathcal{F}}$?

If $\widehat{\mathcal{F}}$ is finite, we say that an algorithm $\mathcal{A}$ for $\widehat{\mathcal{F}}$-SM or $\widehat{\mathcal{F}}$-CM is a polynomial-time algorithm if its running time is polynomial in n, m, and $|\widehat{\mathcal{F}}|$; if $\widehat{\mathcal{F}}$ is infinite, we require the running time of $\mathcal{A}$ to be polynomial in n and m. We remark that $\widehat{\mathcal{F}}$-SM (respectively, $\widehat{\mathcal{F}}$-CM) is in NP for any finite collection $\widehat{\mathcal{F}}$ of polynomially computable voting rules: it suffices to guess a manipulative vote L (respectively, a list $(L_1, \ldots, L_s)$ of manipulative votes) and verify that it makes p the unique winner under every rule in $\widehat{\mathcal{F}}$. Thus, in what follows, when proving that these problems are NP-complete for some finite $\widehat{\mathcal{F}}$, we will only provide an NP-hardness proof.

Traditionally, the problems $\widehat{\mathcal{F}}$-SM and $\widehat{\mathcal{F}}$-CM are studied for the case $|\widehat{\mathcal{F}}| = 1$. In what follows, whenever $\widehat{\mathcal{F}} = \{\mathcal{F}\}$, we omit the curly braces and write $\mathcal{F}$-SM/CM instead of $\{\mathcal{F}\}$-SM/CM to conform with the standard notation. We omit some of the proofs due to space constraints.

4 Manipulation

We start by considering the SM problem. In their classic paper [3], Bartholdi, Tovey and Trick show that this problem is polynomial-time solvable for $\mathsf{Copeland}^{\alpha}$ (for every rational $\alpha \in [0,1]$), Maximin, and all scoring rules (while Bartholdi et al. do not explicitly consider scoring rules other than Plurality and Borda, it is not hard to see that their algorithm works for any scoring rule).

Remarkably, for all these rules the manipulative vote can be found by essentially the same algorithm. This algorithm starts by ranking p first; it is safe to do so, because all of these rules are monotone. Note that at this point we can already compute p's final score; let us denote it by $s(p)$. The algorithm then fills up positions $2,\dots,m$ in the vote one by one. When considering position i, $i \geq 2$, it tries to place each of the still unranked candidates into this position. At this point, the identities of the candidates in positions $1,\dots,i-1$ are already known, so one can determine the score of each candidate c if it were to be placed in position i (this is true for Copeland, Maximin and all scoring rules, but need not be true in general, even for monotone rules); let us denote this quantity by $s_i(c)$. If there exists a candidate c such that $s_i(c) < s(p)$, it is placed in position i; if there are several such candidates, one of them is selected arbitrarily. If no such candidate can be found, the algorithm reports that no manipulative vote exists.

Bartholdi et al. prove the correctness of this algorithm for all voting correspondences that (1) are monotone and (2) have the property that the score of a candidate c can be determined if we know which candidates are ranked above and below c in each vote, and the winners are the candidates with the highest score. $\mathsf{Copeland}^{\alpha}$, $\alpha \in \mathbb{Q} \cap [0,1]$, Maximin, and all scoring rules satisfy both of these conditions, and STV satisfies neither of them; indeed, STV-SM is known to be NP-complete [2].

We will now show that the algorithm of Bartholdi et al. extends to $\widehat{\mathcal{F}}$-SM for any finite set $\widehat{\mathcal{F}}$ that consists of voting correspondences that satisfy (1) and (2).

Theorem 4.1 *Let $\widehat{\mathcal{F}}$ be a finite set of voting rules such that every rule $\mathcal{F}_i \in \widehat{\mathcal{F}}$ satisfies conditions (1) and (2). Then $\widehat{\mathcal{F}}$-SM can be solved in polynomial time.*

The proof of Theorem 4.1 is very simple. However, the result itself plays a key role in our understanding of single-voter manipulation under voting rule uncertainty. Indeed, to the best of our knowledge, for all classic voting rules for which single-voter manipulation is known to be easy, a manipulative vote can be constructed using the algorithm of [3]. Therefore, we cannot hope to put together two or more classic easy-to-manipulate rules so that the manipulation problem with respect to the combination of these rules is computationally hard.

One can nevertheless ask if such a combination of rules exists. We will now show that the answer to this question is "yes": we present two easy-to-manipulate rules, which we will call STV_1 and STV_2, such that STV_i-SM is polynomial-time solvable for $i = 1,2$ but $\{\mathsf{STV}_1, \mathsf{STV}_2\}$-SM is NP-hard. Admittedly, these rules are not particularly natural; but then Theorem 4.1 shows that we cannot hope to prove a result of this type for natural voting rules.

The main idea of the construction is that each of these rules can be manipulated either by making p the STV winner or by using an easy-to-compute "trapdoor"; however, the "trapdoors" for STV_1 and STV_2 are incompatible with each other, so, to manipulate both, one needs to manipulate STV.

Formally, STV_1 is defined as follows. For $m \leq 3$, all candidates are declared to be the winners. For $m > 3$, the rule is not neutral in a very essential way: candidates c_{m-2}, c_{m-1} and c_m play a special role. Specifically, if some voter ranks c_{m-3+j} in position $m-3+j$ for $j = 1,2,3$, then the candidate ranked first by this voter is declared to be the election winner; if there are several such voters, the set of winners consists of these voters' top choices. Otherwise, the winner is the winner under the STV rule.

STV_2 coincides with STV_1 for $m \leq 3$. For $m > 3$, if some voter ranks c_{m-3+j} in position c_{m+1-j} for $j = 1,2,3$, then the candidate ranked first by this voter is declared to be the election winner (again, the election may have multiple winners if there are several such voters), and otherwise the winner is the STV winner.

Theorem 4.2 STV_1-SM *and* STV_2-SM *are in* P. *However,* $\{\mathsf{STV}_1, \mathsf{STV}_2\}$-SM *is* NP-*complete.*

Proof. Consider an instance of STV_1. Suppose that some of the honest voters rank c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$, and let S be the set of these voters' top choices. If $S \neq \{p\}$, no matter what the manipulator does, all candidates in S will be declared the election winners, so the manipulator cannot make p the unique winner. If $S = \{p\}$, or if none of the honest voters ranks c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$, the manipulator can rank p first and place c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$; this would make p the unique winner. In any case, the manipulator's problem is in P. A similar argument shown that STV_2-SM is in P.

To show that $\{\mathsf{STV}_1, \mathsf{STV}_2\}$-SM is NP-hard, we will provide an NP-hardness reduction from STV-SM, which is known to be NP-complete [2].

Given an instance of STV-SM with a set of candidates $C = \{c_1, \dots, c_{m'}\}$, a set of voters V, $|V| = n-1$, a preference profile $\mathcal{R} = (R_1, \dots, R_{n-1})$ over C, and a preferred candidate $p \in C$, we will modify it as follows. We let $m = m' + 3$ and set $C' = C \cup \{c_{m-2}, c_{m-1}, c_m\}$. We ask each of the voters to rank each of the candidates in C in the same position as before, and rank c_{m-1} in position $m-2$, followed by c_{m-2} and c_m; denote the resulting preference profile by $\mathcal{R}'$.

Observe that the manipulator can make p the unique winner of this election under STV_1 either by ranking c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$, or by making p the unique STV winner. Similarly, the manipulator can make p the unique winner of the new election under STV_2 either by ranking c_{m-3+j} in position $m+1-j$ for $j = 1, 2, 3$, or by making p the unique STV winner.

Now, suppose that the original instance of STV-SM is a "yes"-instance, and let $L \in \mathcal{L}(C)$ be the manipulative vote that makes p the STV winner in that election. Consider the vote L' obtained from L by ranking c_{m-1}, c_{m-2}, and c_m after all candidates in C (in this order). In $(\mathcal{R}', L')$, no voter ranks c_{m-2}, c_{m-1}, c_m according to either of the "trapdoors", so both in STV_1 and in STV_2 the STV rule is applied. Further, in $(\mathcal{R}', L')$ candidates c_{m-2}, c_{m-1}, c_m receive no first-place votes, so under STV they are eliminated before any candidates in C. STV then proceeds in the same way as on $(\mathcal{R}, L)$, thus making p the winner.

Conversely, suppose that there exists a vote $L' \in \mathcal{L}(C')$ such that p is the unique winner in $(\mathcal{R}', L')$ with respect to both STV_1 and STV_2. Since L cannot rank c_m in positions $m-2$ and m simultaneously, it follows that p is the STV winner in $(\mathcal{R}', L')$. Now, consider the execution of STV_1 on $(\mathcal{R}', L')$. If L' does not rank any of the candidates in $C' \setminus C$ in the top position, after the first three steps the execution of STV_1 on $(\mathcal{R}', L')$ coincides with the execution of STV on $(\mathcal{R}, L)$, where L is obtained from L' by removing c_{m-2}, c_{m-1} and c_m. Thus, in this case L is a successful manipulative vote that witnesses that the original instance of STV-SM is a "yes"-instance.

Now, suppose that L' ranks a candidate from $C' \setminus C$ first; assume without loss of generality that the top candidate in L is c_m. Then simply removing c_{m-2}, c_{m-1} and c_m from L' would not necessarily work: if the top candidate in the resulting vote receives no first-place votes in $\mathcal{R}$, this candidate would have been eliminated in the very beginning in $(\mathcal{R}', L')$, but may survive much longer in the modified election. Thus, we need a slightly different strategy. Let C_0 be the set of candidates that receive no first-place votes in $\mathcal{R}$. We construct L from L' by removing c_{m-2}, c_{m-1} and c_m and moving candidates in C_0 to the bottom of the vote (without changing the relative ordering of all other candidates). Then on $(\mathcal{R}', L')$ STV starts by eliminating c_{m-1}, c_{m-2} and the candidates in C_0. At this point, each candidate has at least one first-place vote; hence, because of our intermediate tie-breaking rule, c_m is the first candidate to be eliminated, and we are left with an election E'' over $C \setminus C_0$. On the other hand, in $(\mathcal{R}, L)$ the set of candidates with no first-place votes coincides with C_0, so after the first $|C_0|$ elimination rounds we obtain an election over $C \setminus C_0$ that coincides with E''. Hence, p is the unique STV winner in $(\mathcal{R}, L)$, and hence our original instance of STV-SM is a "yes"-instance. ❑

We remark that Theorem 4.2 holds for coalitional manipulation as well: for the easiness result, note that the manipulators may use trapdoors to manipulate STV_1 or STV_2, and the hardness result generalizes trivially.

The next question that we would like to explore is whether a combination of an easy-to-manipulate rule with a hard-to-manipulate one is hard to manipulate. We will now illustrate that this is the case for two classic voting rules, namely, STV and Borda.

Theorem 4.3 $\{\text{Borda}, \text{STV}\}$-SM *is* NP-*complete.*

Proof. We will provide a reduction from STV-SM. Consider an instance of STV-SM given by an election (C, V) with $C = \{c_1, \dots, c_m\}$, $|V| = n - 1$, a preference profile $\mathcal{R} = (R_1, \dots, R_{n-1})$, and a candidate $p \in C$; assume without loss of generality that $n \geq 3$. Suppose that p is not ranked first by any of the voters in V. Then if the manipulator does not rank p first, p get eliminated before any candidate that has a positive Plurality score in (C, V) and therefore does not win the election. Hence, the manipulator has to rank p first. Observe also that the rest of the manipulator's vote does not matter in this case: it can only impact the candidate elimination process after p is eliminated, at which point p has already lost the election. Thus, if no voter in V ranks p first, the manipulator's problem is in P: the manipulator should rank p first and check if this achieves the desired result. We can therefore assume without loss of generality that in our input instance of STV-SM candidate p receives at least one first-place vote.

Thus, assume that p is the top candidate of voter 1. Let $D = \{c_{im+j} \mid i = 1, \dots, n, j = 1, \dots, m\}$, and set $C' = C \cup D$. Modify all votes in $\mathcal{R}$ by inserting the candidates in D right below p in each vote, in an arbitrary order; let $\mathcal{R}'$ be the resulting profile.

Let $s(c)$ denote the Borda score of a candidate $c \in C$ in $(C, V, \mathcal{R})$, and let $s'(c)$ denote his score in $(C', V, \mathcal{R}')$. We have $s(c) \leq (n-1)(m-1)$ for all $c \in C$. Moreover, we have $s'(p) = s(p) + mn(n-1)$, as p gets mn extra points from each vote. On the other hand, every other candidate in C gets at most $mn(n-2)$ extra points from voters $2, \dots, n-1$ and no extra points from voter 1. Thus, for any $c \in C \setminus \{p\}$ we have

$$s'(c) \leq s(c) + mn(n-2) \leq mn(n-1) - m - n + 1 < s'(p) - m.$$

Also, the Borda score of any $d \in D$ in $(C', V, \mathcal{R}')$ is less than $s'(p)$. Thus, if the manipulator ranks the candidates in C in top m positions, p is the unique Borda winner of the resulting election.

On the other hand, no matter how the manipulator votes, under STV all candidates in D will be eliminated before all candidates in C that have a non-zero Plurality score: indeed, the Plurality score of each $d \in D$ is at most 1, and the intermediate tie-breaking rule favors candidates in C over those in D.

We are now ready to show that our reduction is correct. Let L be a successful manipulative vote for the original instance, and let C_0 be the set of all candidates in C with no first-place votes in $(\mathcal{R}, L)$. Note that the candidates in C_0 are eliminated in the first $|C_0|$ rounds of STV. Now, consider the vote L' obtained from L by ranking the candidates in D in positions $m+1, \dots, m(n+1)$. In the election $(\mathcal{R}', L')$ candidate p has the highest Borda score. Moreover, under STV we will first eliminate all candidates in $C_0 \cup D$. At this point, we obtain the same election as after $|C_0|$ rounds of STV on $(\mathcal{R}, L)$—and hence the same winner. Thus, L' is a successful manipulative vote in the new election.

Conversely, suppose that $L' \in \mathcal{L}(C \cup D)$ is such that in $(\mathcal{R}', L')$ candidate p is both the unique Borda winner and the (unique) STV winner. Let C'_0 be the set of candidates in C that have no first-place votes in $(\mathcal{R}', L')$. When we execute STV on $(\mathcal{R}', L')$, we eliminate all candidates in $D \cup C'_0$ prior to eliminating any of the candidates in $C \setminus C'_0$. Let L be the vote in $\mathcal{L}(C)$ obtained by deleting all candidates in D from L' and moving all candidates in C'_0 to the bottom $|C'_0|$ positions (without changing the relative ordering of the candidates in $C \setminus C'_0$). Then C'_0 is exactly the set of candidates in C who have no first-place votes in $(\mathcal{R}, L)$. Therefore, when we execute STV on $(\mathcal{R}, L)$, we eliminate all candidates in C'_0 prior to eliminating any candidates in $C \setminus C'_0$. Thus, the profile obtained after running STV for $|D| + |C'_0|$ steps on $(\mathcal{R}', L')$ coincides with the profile obtained after running STV for $|C'_0|$ steps on $(\mathcal{R}, L)$. Thus, L is a successful manipulative vote for the original election. ❑

Another interesting (and arguably natural) combination of voting rules is $\{\mathsf{Plurality}, \mathsf{STV}\}$. Here, we were unable to provide a black-box reduction showing that the combination of these rules is hard to manipulate. However, a careful inspection of Bartholdi and Orlin's proof [2] establishes that $\{\mathsf{Plurality}, \mathsf{STV}\}$-SM is indeed NP-hard: by tweaking the instance of STV constructed in that proof we can ensure that the manipulator's preferred candidate is the unique Plurality winner.

However, there are also examples where the combination of a hard-to-manipulate rule and an easy-to-manipulate one is easy to manipulate. Consider, for instance, the following rule: if some candidate receives strictly more than $\lfloor n/2 \rfloor$ first-place votes, he is the unique election winner; otherwise, all candidates are winners. We will refer to this rule as the Majority rule. Majority is not particularly decisive, but apart from that it is a reasonable voting rule. Clearly, it is easy to manipulate: the manipulator simply needs to check if ranking p first does the job. Moreover, the combination of Majority and STV is easy to manipulate, too.

Theorem 4.4 $\{\mathsf{Majority}, \mathsf{STV}\}$-SM *is in* P.

Proof. Consider an election $E = (C, V, \mathcal{R})$. If in this election p is ranked first by at most $\lfloor n/2 \rfloor - 1$ voters, the manipulator cannot make p the Majority winner, so this is a "no"-instance of our problem. On the other hand, if p is ranked first by at least $\lfloor n/2 \rfloor$ voters, the manipulator can rank p first, making him both the unique Majority winner and the unique STV winner. ❑

The reason why the combination of Majority and STV is easy to manipulate is that Majority is always guaranteed to elect the STV winner: if some candidate has more than $\lfloor n/2 \rfloor$ votes, he will obviously win under STV, and in all other cases Majority elects all candidates. Using this observation, we can now generalize Theorem 4.4. We will say that a voting correspondence $\mathcal{F}_1$ is a *refinement* of a voting correspondence $\mathcal{F}_2$ if for any election E we have $\mathcal{F}_1(E) \subseteq \mathcal{F}_2(E)$, and there exists an election for which this containment is strict. Now, it is easy to see that STV is a refinement of Majority. Also, some of the voting rules defined in Section 2 are refinements of each other: namely, both Copeland and Maximin are refinements of Condorcet. Yet another example is provided by the so-called second-order Copeland rule, proved to be NP-hard to manipulate in [3]: this rule is obtained by combining the Copeland rule with a rather sophisticated tie-breaking rule, and is therefore a refinement of Copeland. Now, it is easy to see that the proof of Theorem 4.4 implies a more general fact.

Corollary 4.5 *If a voting correspondence* $\mathcal{F}_1$ *is a refinement of a voting correspondence* $\mathcal{F}_2$ *and* $\mathcal{F}_2$-SM *is in* P*, then so is* $\{\mathcal{F}_1, \mathcal{F}_2\}$-SM.

We remark that Corollary 4.5 crucially relies on the fact that we consider the unique-winner version of SM, and the requirement that a voting correspondence should produce a non-empty set of winners for every election. Also, the converse of Corollary 4.5 is not true, as illustrated by Copeland and second-order Copeland. Another important observation is that Corollary 4.5 applies equally well to the coalitional manipulation problem; we will make use of this fact in Section 5.

Now, suppose we have two hard-to manipulate rules. Clearly, it can be the case that their combination is also hard to manipulate: for example we can take two copies of STV (if we insist that these two rules should be distinct, we can modify one of the copies to produce a different winner on a single profile; this does not affect the complexity of our problem). To conclude this section, we provide an example of two voting rules $\mathcal{F}_1$ and $\mathcal{F}_2$ such that both $\mathcal{F}_1$-SM and $\mathcal{F}_2$-SM are NP-complete, but $\{\mathcal{F}_1, \mathcal{F}_2\}$-SM is in P; thus, counterintuitively, even a combination of hard-to-manipulate rules can be "easy" to manipulate (it will become clear in a minute why we used quotes in the previous sentence).

Our first voting rule is STV. Our second rule, which we will denote by STV$'$, is obtained from STV by the following modification: if c_i is the STV winner in E, then we output c_{i+1} as the unique winner (where $c_{m+1} := c_1$). Now, clearly, manipulating STV$'$ is just as hard as manipulating STV: we simply have to solve the STV manipulation problem for a different candidate. However, for any election E, STV and STV$'$ have different winners, so there is no way the manipulator can make p win under both of them. Thus, the manipulator's problem is "easy", in the sense that it simply cannot achieve its goal, so every instance of $\{\text{STV}, \text{STV}'\}$-SM is a "no"-instance. We summarize these observations as follows.

Theorem 4.6 STV$'$-SM *is* NP*-complete. On the other hand,* $\{\text{STV}, \text{STV}'\}$-SM *is in* P.

We remark that Theorem 4.6 extends trivially to coalitional manipulation.

5 Coalitional Manipulation

The coalitional manipulation problem is known to be NP-hard for many prominent voting rules, such as Borda [6, 8] and some other scoring rules [19], Copeland$^\alpha$ for $\alpha \in (\mathbb{Q} \cap [0, 1]) \setminus \{0.5\}$ [11, 12] and Maximin [20]; it goes without saying that the hardness result for STV-SM [2] implies that STV-CM is NP-hard as well. Therefore, we cannot hope for a general easiness result along the lines of Theorem 4.1. Nevertheless, we can identify some interesting combinations of voting rules for which CM is in P.

We start by observing that Condorcet-CM is in P. Indeed, the manipulators can simply rank p first in all of their votes and check if that makes p the Condorcet winner; note that the answer to this question does not depend on how the manipulators rank the other candidates. Now, by extending Corollary 4.5 to the coalitional manipulation problem, and using the fact both Maximin and Copeland are refinements of the Condorcet rule, we obtain the following corollaries.

Corollary 5.1 $\{\text{Condorcet}, \text{Maximin}\}$-CM *is in* P.

Corollary 5.2 $\{\text{Condorcet}, \text{Copeland}^\alpha\}$-CM *is in* P *for any* $\alpha \in \mathbb{Q} \cap [0, 1]$.

Of course, the coalitional manipulation problem is also easy for the Majority rule, and it can be easily checked that each of the rules defined in Section 2 is a refinement of the Majority rule. Thus, we could obtain a similar easiness result for the combination of Majority and any other rule. We chose to state Corollaries 5.1 and 5.2 for the Condorcet rule, as the latter is more decisive and has been considered in prior work on computational social choice, albeit in the context of control [4].

We will now move on to another family of voting rules whose combinations can be shown to be easy to manipulate. A recent paper by Lin [17] shows that the coalitional manipulation problem is easy for k-Approval for any value of k. We will now prove a stronger statement: coalitional manipulation is easy even for combinations of k-Approval rules (for different values of k).

Theorem 5.3 *For any finite set* $K = \{k_1, \dots, k_\ell\} \subseteq \mathbb{N}$*, the problem* $\{k_1\text{-Approval}, \dots, k_\ell\text{-Approval}\}$-CM *is in* P.

Proof. Consider an election E with $C = \{c_1, \dots, c_m\}$, $|V| = h$, $|M| = s$, and $\mathcal{R} = (R_1, \dots, R_h)$. We can assume without loss of generality that $p = c_m$.

Since k-Approval is monotone for any value of k, it is optimal for the manipulators to rank p first in all s votes. For each $k \in K$, let $s_k(p)$ be p's k-Approval score in the resulting election. Now, the manipulators' goal is to rank every other candidate $c \in C \setminus \{p\}$ so that for each $k \in K$ the k-Approval score of c is strictly less than $s_k(p)$. We can assume without loss of generality that for each $k \in K$ and each $c \in C \setminus \{p\}$ the k-Approval score of c in $\mathcal{R}$ is strictly less than $s_k(p)$: otherwise, we clearly have a "no"-instance of our problem. Now, for each $r = 2, \dots, m$ and each c_j, $j = 1, \dots m - 1$, let $x(r, j)$ be the maximum number of times that c_j can be ranked in position r or higher in the manipulators' votes so that its k-Approval score is less than $s_k(p)$ for every $k \in K$. These values are easy to compute from the candidates' k-Approval scores in $\mathcal{R}$, $k \in K$; our assumption on the initial scores ensures that they are non-negative.

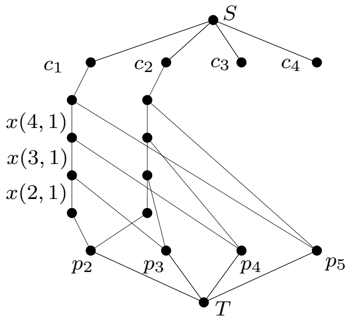


Figure 1: Network in the proof of Theorem 5.3, $m = 5$

We will now construct a flow network so that the maximum flow in this network corresponds to a successful set of manipulative votes, if one exists. Our network has a source S, a sink T, a node c_j for each $j = 1, \ldots, m-1$, and a node p_r for $r = 2, \ldots, m$; intuitively, node p_r corresponds to position r in the manipulators' votes. There is an edge of capacity s from S to each c_j, $j = 1, \ldots, m-1$, and an edge of capacity s from each p_r, $r = 2, \ldots, m$, to T. Essentially, the edge from S to c_j ensures that c_j is ranked by each manipulator, and the edge from p_r to T ensures that each of the manipulators fills position r in his vote. It remains to explain how to connect the candidates with the positions.

For each $c_j \in C \setminus \{p\}$ we build a caterpillar graph that connects c_j to $p_m, \ldots, p_2$. More formally, for each candidate c_j, $j = 1, \ldots, m-1$, we introduce nodes $z_{j,m}, \ldots, z_{j,2}$ and edges $(c_j, z_{j,m})$, $(z_{j,r}, z_{j,r-1})$ for $r = m, \ldots, 3$, and $(z_{j,r}, p_r)$ for $r = m, \ldots, 2$. The capacity of $(c_j, z_{j,m})$ and $(z_{j,r}, p_r)$, $r = m, \ldots, 2$, is $+\infty$, and the capacity of $(z_{j,r}, z_{j,r-1})$, $r = m, \ldots, 3$, is given by $x(r-1, j)$. This completes the description of our network (see Figure 1).

We claim that this network admits a flow of size $s(m-1)$ if and only if there exists an assignment of candidates to the positions in the manipulators' votes such that the k-Approval score of each $c \in C \setminus \{p\}$ is less than $s_k(p)$ for every $k \in K$. Indeed, suppose that such a flow exists. Since all capacities are integer, we can assume that this flow is integer. It saturates all edges leaving S, so there are s units of flow leaving each c_j, $j = 2, \ldots, m$. This flow has to reach $p_2, \ldots, p_m$ traveling through the caterpillar graph associated with c_j. Thus, we can associate the flow on the edge $(z_{j,r}, p_r)$ with the number of times that c_j is ranked in position p_r. The capacity constraints on edges guarantee that these numbers correspond to a valid set of manipulators' votes. Moreover, for each $r = m, \ldots, 2$, the total flow from c_j to $p_r, \ldots, p_2$ is at most $x(r, j)$, which ensures that c_j is ranked in positions $p_r, \ldots, p_2$ at most $x(r, j)$ times. Hence, for each $k \in K$ and each $j = 1, \ldots, m-1$, the k-Approval score of c_j is less than that of p, and therefore p is the unique winner under each of the rules in our collection. Conversely, a vote that makes p the unique election winner with respect to each k-Approval, $k \in K$, can be converted into a valid flow; if x manipulators rank c_j in position r, we send x units of flow on $(z_{j,r}, p_r)$. ❑

Theorem 5.3 has an interesting implication. Let $\widehat{\mathcal{F}}_\alpha$ be the family of all scoring rules. Observe that $\widehat{\mathcal{F}}_\alpha$ includes the Borda rule, for which coalitional manipulation is hard. Nevertheless, it turns out that $\widehat{\mathcal{F}}_\alpha$-CM is solvable in polynomial time.

Theorem 5.4 $\widehat{\mathcal{F}}_\alpha$-CM *is in* P.

We will now provide several examples of combinations of rules for which coalitional manipulation is hard. We will focus on classic voting rules, and investigate combinations of the most prominent easy-to-manipulate rule, namely, Plurality, with Borda and Copeland, which are both hard for coalitional manipulation.

Theorem 5.5 {Plurality, Borda}-CM *is* NP-*complete.*

It is interesting to compare Theorem 5.4 and Theorem 5.5: the former implies that the combination of Borda with *all* k-Approval rules is easy to manipulate, whereas the latter shows that the combination of 1-Approval (i.e., Plurality) and Borda is hard to manipulate; we remark that the proof of Theorem 5.5 extends easily to the combination of Borda with k-Approval for any constant k.

A construction similar to the one used in the proof of Theorem 5.5 shows that $\{\mathsf{Plurality}, \mathsf{Copeland}^\alpha\}$-CM is NP-complete for $\alpha \in (\mathbb{Q} \cap [0,1]) \setminus \{0.5\}$ (this is the range of values of α for which $\mathsf{Copeland}^\alpha$-CM in known to be NP-complete). The only difference is that for Copeland we cannot assume that the number of voters is a small constant (we will, however, assume that there are exactly two manipulators, as this is known to be sufficient for the NP-hardness of this problem [11, 12]). Therefore, instead of adding one pair (X_i, X_i') for each $i = 1, \dots, m-1$, we add h such pairs, where h is the number of honest voters. This modification has no impact on Copeland scores: if c beats d in the original profile, this remains to be the case when the new votes are added; the converse is also true. However, the Plurality score of p increases by $h(m-1)$, whereas the Plurality score of any other candidate increases by h, and, as a result, does not exceed $2h+2$ (even taking the manipulators' votes into account). Assuming without loss of generality that $m \geq 4$ and $h \geq 3$, we obtain that p is the unique Plurality winner of the modified election, irrespective of how the manipulator votes. The rest of the argument proceeds as in the proof of Theorem 5.5. We obtain the following corollary.

Corollary 5.6 $\{\mathsf{Plurality}, \mathsf{Copeland}^\alpha\}$-CM *is* NP*-complete for* $\alpha \in (\mathbb{Q} \cap [0,1]) \setminus \{0.5\}$.

Perhaps unsurprisingly, the combination of Borda and Copeland is hard to manipulate as well.

Theorem 5.7 $\{\mathsf{Borda}, \mathsf{Copeland}^\alpha\}$-CM *is* NP*-complete for* $\alpha \in (\mathbb{Q} \cap [0,1]) \setminus \{0.5\}$.

We remark that the proofs of Theorems 4.3, 5.7 and 5.5 and Corollary 5.6 are based on the same idea: we can modify an election so that the (relative) scores of all candidates with respect to one rule remain essentially unchanged while making a certain candidate a winner with respect to another voting rule. This suggests that these rules exhibit certain independence; this is somewhat reminiscent of Klamler's work on closeness of voting rules (see Klamler [15] and references therein). Formalizing this notion of independence is an interesting direction for future work.

6 Conclusions and Future Work

We have investigated the problem of (coalitional) manipulation under uncertainty about the voting rules. Our results are summarized in Table 1.

	SM	CM
easy + easy = easy	all "nice" rules	k-Approval
easy + easy = hard	$\{\mathsf{STV}_1, \mathsf{STV}_2\}$	$\{\mathsf{STV}_1, \mathsf{STV}_2\}$
easy + hard = hard	$\{\mathsf{Borda}, \mathsf{STV}\}$, $\{\mathsf{Plurality}, \mathsf{STV}\}$	$\{\mathsf{Plurality}, \mathsf{Borda}\}$, $\{\mathsf{Plurality}, \mathsf{Copeland}\}$
easy + hard = easy	$\{\mathsf{Majority}, \mathsf{STV}\}$	$\{\mathsf{Condorcet}, \mathsf{Copeland}\}$, $\{\mathsf{Condorcet}, \mathsf{Maximin}\}$, scoring rules
hard + hard = easy	$\{\mathsf{STV}, \mathsf{STV}'\}$	$\{\mathsf{STV}, \mathsf{STV}'\}$
hard + hard = hard	$\{\mathsf{STV}, \mathsf{STV}\}$	$\{\mathsf{Borda}, \mathsf{Copeland}\}$

Table 1: Summary of results

While we have not established the complexity of our problem for all possible combinations of voting rules, our results identify a number of approaches for dealing with problems of this type and the features of voting rules that make their combinations easy or hard to manipulate.

An obvious direction for future work is extending our approach to other forms of cheating in elections, such as control and bribery. Also, an interesting variant of our problem in the context of single-winner manipulation can be obtained by adopting the paradigm of safe strategic voting [18]. That is, instead of assuming that the manipulator wants to get a certain candidate elected, we take the more traditional approach, where the manipulator, too, has a preference order and would like to improve the election outcome with respect to this order; we can then ask whether the manipulator can vote so that the outcome improves for at least one voting rule in the given family and does not get worse with respect to the other rules.

References

[1] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *AAAI'10*, pages 697–702, 2010.

[2] J. Bartholdi and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[3] J. Bartholdi, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[4] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[5] D. Baumeister, M. Roos, and J. Rothe. Computational complexity of two variants of the possible winner problem. In *AAMAS'11*, pages 853–860, 2011.

[6] N. Betzler, R. Niedermeier, and G. J. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *IJCAI'11*, pages 55–60, 2011.

[7] V. Conitzer, T. Walsh, and L. Xia. Dominating manipulations in voting with partial information. In *AAAI'11*, pages 638–643, 2011.

[8] J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. Complexity of and algorithms for Borda manipulation. In *AAAI'11*, pages 657–662, 2011.

[9] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *AAAI'10*, pages 768–773, 2010.

[10] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *ISAAC'05*, pages 206–215, 2005.

[11] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *AAMAS'08*, pages 983–990, 2008.

[12] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Manipulation of Copeland elections. In *AAMAS'10*, pages 367–374, 2010.

[13] P. Faliszewski and A. D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[14] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

[15] C. Klamler. On the closeness aspect of three voting rules: Borda—Copeland—Maximin. *Group Decision and Negotiation*, 14:233–240, 2005.

[16] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *MPREF'05*, pages 124–129, 2005.

[17] A. Lin. The complexity of manipulating k-approval elections. In *ICAART'11*, pages 212–218, 2011.

[18] A. Slinko and S. White. Non-dictatorial social choice rules are safely manipulable. In *COMSOC'08*, pages 403–413, 2008.

[19] L. Xia, V. Conitzer, and A. Procaccia. A scheduling approach to coalitional manipulation. In *ACM EC'10*, pages 275–284, 2010.

[20] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer, and J. Rosenschein. Complexity of unweighted manipulation under some common voting rules. In *IJCAI'09*, pages 348–353, 2009.

Edith Elkind
Division of Mathematical Sciences
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
Email: eelkind@ntu.edu.sg

Gábor Erdélyi
Institut für Betriebswirtschaftslehre
Universität Siegen
Siegen, Germany
Email: erdelyi@wiwi.uni-siegen.de

The Complexity of Nearly Single-Peaked Consistency[1]

Gábor Erdélyi, Martin Lackner, and Andreas Pfandler

Abstract

Manipulation, bribery, and control are well-studied ways of changing the outcome of an election. Many voting systems are in the general case computationally resistant to some of these manipulative actions. However when restricted to single-peaked electorates, these problems suddenly become easy to solve. Recently, Faliszewski, Hemaspaandra, and Hemaspaandra [FHH11] studied the complexity of dishonest behavior in nearly single-peaked electorates. These are electorates that are not single-peaked but close to it according to some distance measure.

In this paper we introduce several new distance measures regarding single-peakedness. We prove that determining whether a given profile is nearly single-peaked is in many cases NP-complete. Furthermore, we explore the relations between several notions of nearly single-peakedness.

1 Introduction

Voting is a very useful method for preference aggregation and collective decision-making. It has applications in very broad settings ranging from politics to artificial intelligence and further topics in computer science (see, e.g., [DKNS01, ER97, GMHS99]). In the presence of huge data volumes, the computational properties of voting rules are worth studying. In particular, we usually want to determine the winners of an election quickly. On the other hand we want to make various forms of dishonest behavior computationally as hard as possible.

Bartholdi, Tovey, and Trick [BTT89] were the first to study the computational aspects of manipulation in elections, where a group of voters cast their votes insincerely in order to reach a desired outcome. Other types of dishonest behavior are control, where an external agent makes structural changes on the election such as adding/deleting/partitioning either candidates or voters (see, e.g., [BTT92]) in order to reach a desired outcome, or bribery, where an external agent changes some voters' votes in order to change the outcome of the election (see, e.g., [FHH09]). For an overview and many natural examples on bribery, control, and manipulation we refer to the survey of Baumeister et al. [BEH$^+$10].

Traditionally, the complexity of such attacks on the outcome is studied under the assumption that in each election any admissible vote can occur. However, there are many elections where the diversity of the votes is limited in a sense that there are some admissible votes nobody would ever cast. One of the best known examples is *single-peakedness*, introduced by Black [Bla48]. It assumes that the votes are polarized along some linear axis. The study of the computational aspects of elections with single-peaked preferences was initiated by Walsh [Wal07] (see also [FHHR11, BBHH10]). In many cases NP-hardness results from the general cases turn out to be easy in single-peaked societies. A recent line of research initiated by Conitzer [Con09] and by Escoffier, Lang, and Öztürk [ELÖ08] suggests that many elections are not perfectly single-peaked but are *close* to it with respect to some metric. Faliszewski, Hemaspaandra, and Hemaspaandra [FHH11] introduced various notions of nearly single-peaked elections and showed that the complexity of manipulative-actions jumps back to NP-hardness in many cases.

In this paper we consider the notion of k-maverick single-peakedness and k-local swaps introduced by Faliszewski, Hemaspaandra, and Hemaspaandra [FHH11]. In addition we follow the sug-

[1]This work was done in part while the second and the third authors were visiting Universität Siegen and while the first author was visiting Vienna University of Technology. The work of the second and third author was supported by the Austrian Science Fund (FWF): P20704-N18.

gestions of Escoffier, Lang, and Öztürk [ELÖ08] and formally define the two nearly single-peaked notions k-candidate deletion and k-additional axes. Furthermore, we introduce three new notions of nearly single-peakedness, k-local candidate deletion, k-global swaps, and k-candidate partition. We show connections between the existing and new notions, and we study the complexity of determining whether a given profile is nearly single-peaked with respect to some axis. This problem was introduced by Escoffier, Lang, and Öztürk [ELÖ08] as *single-peaked consistency*. We show that single-peaked consistency is computationally hard for four notions of nearly single-peakedness given in this paper. The complexity of the remaining three notions is still open.

Related Work Our paper fits in the line of research on single-peaked and nearly single-peaked preferences. Faliszewski et al. [FHHR11] and Brandt et al. [BBHH10] investigate the complexity of dishonest behavior (e.g., the complexity of manipulation and control) in electorates with single-peaked preferences as well as the winner problem. They do not consider nearly single-peaked preferences, but mention them as future work.

In the context of nearly single-peaked preferences the most relevant paper is by Faliszewski, Hemaspaandra, and Hemaspaandra [FHH11]. They introduce several notions of nearly single-peakedness and analyze the complexity of bribery, control, and manipulation under those conditions. In contrast, we are not analyzing dishonest behavior in elections, but we are studying the complexity of nearly single-peaked consistency.

The question whether a given profile is single-peaked has been recently investigated by Escoffier, Lang, and Öztürk [ELÖ08]. The difference in their work is that they have not considered nearly single-peakedness but they only pointed it out as a possible future research direction.

The idea of measuring the distance of votes with the number of required swaps required to make them identical already appears in Dodgon's voting rule (see, e.g., [MN08] for a discussion). This idea has been widely used since then. Elkind, Faliszewski, and Slinko used swaps of adjacent candidates in votes in the context of bribery [EFS09]. They assumed that a briber can perform a number of swaps in the votes in order to make his favourite candidate win the election. In our paper, we use swaps as a distance measure for nearly single-peakedness. We do not want to change the outcome of an election, we just want to measure the swap distance of a given profile to the nearest single-peaked profile.

Finally, we remark that single-peaked preferences have been considered in the context of preference elicitation [Con09] and in the context of possible and necessary winners under uncertainty regarding the votes [Wal07].

Organization This paper is organized as follows. In Section 2, we recall some notions from voting theory and define single-peaked preferences. In Section 3, we introduce the problems we are investigating in our paper. Our results on the relations between the different notions of single-peakedness and on the complexity of single-peaked consistency are presented in Section 4. Finally, Section 5 provides some conclusions and future directions.

2 Preliminaries

Let C be a finite set of *candidates*, V be a finite set of *voters*, and let $\succ$ be a *preference relation* (i.e., a tie-free and total order) over C. We call a candidate c the *peak* of a preference relation $\succ$ if $c \succ c_i$ for all $c_i \in C \setminus \{c\}$. Let $\mathscr{P} = (\succ_1, \dots, \succ_n)$ be a *preference profile* (i.e., a collection of linear orders) over the candidate set C. We say that the preference order $\succ_i$ is the *vote* of voter i. For simplicity, we will write for each voter $i \in V$ $c_1 c_2 \dots c_n$ instead of $c_1 \succ_i c_2 \succ_i \dots \succ_i c_n$. We call the peak of voter i his highest ranked or *top-ranked* candidate. An *election* is defined as a triple $E = (C, V, \mathscr{P})$, where C is the set of candidates, V the set of voters and $\mathscr{P}$ a preference profile over C.

In order to define single-peaked profiles we will make use of the definition given by Escoffier et al. [ELÖ08].

Definition 2.1 ([ELÖ08])**.** *Let an* axis *A be a total order over C denoted by $>$. Given two candidates $c_i, c_j \in C$, a vote $k \in V$ specified by the corresponding preference relation $\succ_k$, and an axis A. Let c be the top-ranked candidate of voter k. We say that candidates c_i and c_j are* on the same side of the peak of $\succ_k$ *if one of the following two conditions holds:*

(1) $c_i > c$ *and* $c_j > c$, *or* *(2)* $c > c_i$ *and* $c > c_j$

A vote k is said to be single-peaked with respect to an axis A *if for all $c_i, c_j \in C$ that are on the same side of the peak c of $\succ_k$ it holds that $c_i \succ_k c_j$ if either $c > c_i > c_j$ or $c_j > c_i > c$ holds (i.e., c_i is closer to the peak than c_j).*

A preference profile $\mathscr{P}$ is said to be single-peaked with respect to an axis A *if and only if each vote is single-peaked with respect to A. A preference profile $\mathscr{P}$ is said to be* single-peaked consistent *if there is an axis A such that $\mathscr{P}$ is single-peaked with respect to A.*

Let $C' \subseteq C$. By $\mathscr{P}[C']$ we denote the profile $\mathscr{P}$ restricted to the candidates in C'. Analogously if A is an axis over C, we denote by $A[C']$ the axis A restricted to candidates in C'.

Escoffier, Lang, and Öztürk present an algorithm that decides whether a given preference profile is single-peaked consistent in time $|V| \cdot |C|$ [ELÖ08]. Their algorithm improves upon the runtime of an algorithm presented in [BT86]. The corresponding decision problem is defined as follows.

SINGLE-PEAKED CONSISTENCY	
Given:	An election $E = (C, V, \mathscr{P})$.
Question:	Is $\mathscr{P}$ single-peaked consistent?

3 Problem Statement

In this paper we consider different notions of nearly single-peakedness. All these notions define a distance measure to single-peaked profiles. We will now describe them and provide first (trivial) upper bounds on these distances.

k-Maverick

The first formal definition of nearly single-peaked societies was given by Faliszewski, Hemaspaandra, and Hemaspaandra [FHH11]. Consider a preference profile $\mathscr{P}$ for which most voters are single-peaked with respect to some axis A. All voters that are not single-peaked with respect to A are called mavericks. The number of mavericks defines a natural distance measure to single-peakedness. If an axis can be found for a large subset of the voters, this is still a fundamental observation about the structure of the votes.

Definition 3.1 ([FHH11])**.** *Let $E = (C, V, \mathscr{P})$ be an election and k a positive integer. We say that the profile $\mathscr{P}$ is* k-maverick single-peaked consistent *if by removing at most k preference relations (votes) from $\mathscr{P}$ one can obtain a preference profile $\mathscr{P}'$ that is single-peaked consistent.*

Let $M(\mathscr{P})$ denote the smallest k such that $\mathscr{P}$ is k-maverick single-peaked consistent. Note that $M(\mathscr{P}) \leq |V| - 1$ always holds.

The above notion is a well-motivated distance regarding single-peakedness, but we will define other distances which could be more useful in other cases.

k-Candidate Deletion

As suggested in [ELÖ08], we introduce outlier candidates. These are candidates that do not have "a correct place" on any axis and consequently have to be deleted in order to obtain a single-peaked consistent profile. Examples could be a candidate that is not well-known (e.g., a new political party) or a candidate that prioritizes other topics than most candidates and thereby is judged by the voters according to different criteria. The votes restricted to the remaining candidates might still have a clear and significant structure, i.e., might be single-peaked consistent.

Definition 3.2. *Let $E = (C,V,\mathscr{P})$ be an election and k a positive integer. We say that the profile $\mathscr{P}$ is k-*candidate deletion single-peaked consistent *if we can obtain a set $C' \subseteq C$ by removing at most k candidates from C such that the preference profile $\mathscr{P}[C']$ is single-peaked consistent.*

Let $CD(\mathscr{P})$ denote the smallest k such that $\mathscr{P}$ is k-candidate deletion single-peaked consistent. Note that $CD(\mathscr{P}) \leq |C|-2$ always holds.

k-Local Candidate Deletion

Personal friendships or hatreds between voters and candidates could move candidates up or down in a vote. These personal relationships cannot be reflected in a global axis. To eliminate the influence of personal relationships to some candidates we define a local version of the previous notion. This notion can also deal with the possibility that the least favourite candidates might be ranked without special consideration or even randomly.

We first have to define partial domains and partial profiles.

Definition 3.3. *Let C be a set of candidates and A an axis over C. A preference relation $\succ$ over a candidate set $C' \subset C$ is called a partial vote. It is said to be* single-peaked with respect to A *if it is single-peaked with respect to $A[C']$. A partial preference profile consists of partial votes. It is called* single-peaked consistent *if there exists an axis A such that its partial votes are single-peaked with respect to A.*

Definition 3.4. *Let $E = (C,V,\mathscr{P})$ be an election and k a positive integer. We say that the profile $\mathscr{P}$ is k-*local candidate deletion single-peaked consistent *if by removing at most k candidates from each vote in V we obtain a partial preference profile $\mathscr{P}'$ that is single-peaked consistent.*

Let $LCD(\mathscr{P})$ denote the smallest k such that $\mathscr{P}$ is k-local candidate deletion single-peaked consistent. Note that $LCD(\mathscr{P}) \leq |C|-2$ always holds.

k-Additional Axes

Another suggestion in [ELÖ08] is to consider the minimum number of axes such that each preference relation of the profile is single-peaked with respect to at least one of these axes. This notion is particularly useful if each candidate represents opinions on several issues (as it is the case in political elections). A voter's ranking of the candidates would then depend on which issue is considered most important by the voter and consequently each issue might give rise to its own corresponding axis.

Definition 3.5. *Let $E = (C,V,\mathscr{P})$ be an election and k a positive integer. We say that the profile $\mathscr{P}$ is k-*additional axes single-peaked consistent *if there is a partition $V_1,\dots,V_{k+1}$ of V such that the corresponding preference profiles $\mathscr{P}_1,\dots,\mathscr{P}_{k+1}$ are single-peaked consistent.*

Let $AA(\mathscr{P})$ denote the smallest k such that $\mathscr{P}$ is k-additional axes single-peaked consistent. Note that $AA(\mathscr{P}) < \min\left(|V|, \frac{|C|!}{2}\right)$ always holds. This is because the number of distinct votes is trivially bounded by $|V|$. Furthermore, $AA(\mathscr{P})$ is bounded by $\frac{|C|!}{2}$ since at most $|C|!$ distinct votes exist and each vote and its reverse are single-peaked with respect to the same axes.

k-Global Swaps

There is a second method of dealing with candidates that are "not placed correctly" according to an axis A. Instead of deleting them from either the candidate set C or from a vote, we could try to move them to the right position. We do this by performing a sequence of swaps of consecutive candidates. For example, to get from vote *abcd* to vote *adbc*, we first have to swap candidates c and d, and then we have to swap b and d. Since this changes the votes in a more subtle way, this can be considered a less obtrusive notion than k-(Local) Candidate Deletion.

Definition 3.6. *Let $E = (C, V, \mathscr{P})$ be an election and k a positive integer. We say that the profile $\mathscr{P}$ is k-*global swaps single-peaked consistent *if $\mathscr{P}$ can be made single-peaked by performing at most k swaps in the profile. (Note that these swaps can be performed wherever we want – we can have k swaps in only one vote, or one swap each in k votes.)*

Let $GS(\mathscr{P})$ denote the smallest k such that $\mathscr{P}$ is k-global swaps single-peaked consistent. Note that $GS(\mathscr{P}) \leq \binom{|C|}{2} \cdot |V|$ always holds since rearranging a total order in order to obtain any other total order requires at most $\binom{|C|}{2}$ swaps.

k-Local Swaps

We can also consider a "local budget" for swaps, i.e., we allow up to k swaps per vote. This distance measure has been introduced in [FHH11] as Dodgson_k.

Definition 3.7. *Let $E = (C, V, \mathscr{P})$ be an election and k a positive integer. We say that the profile $\mathscr{P}$ is k-*local swaps single-peaked consistent *if $\mathscr{P}$ can be made single-peaked consistent by performing no more than k swaps per vote.*

Let $LS(\mathscr{P})$ denote the smallest k such that $\mathscr{P}$ is k-local swaps single-peaked consistent. Note that $LS(\mathscr{P}) \leq \binom{|C|}{2}$ always holds.

k-Candidate Partition

Our last nearly single-peaked formalism is the candidate analogon of k-additional axes. In this case we partition the set of candidates into subsets such that all of the restricted profiles are single-peaked consistent. This notion is useful in the following situation. Each candidate has an opinion on a controversial Yes/No-issue. Depending on their own preference voters will always rank all Yes-candidates before or after all No-candidates. It might be that when considering only the Yes- respectively No-candidates, the election is single-peaked. Therefore, if we acknowledge the importance of this Yes/No-issue and partition the candidates accordingly, we may obtain two single-peaked elections.

Definition 3.8. *Let $E = (C, V, \mathscr{P})$ be an election and k a positive integer. We say that the profile $\mathscr{P}$ is k-*candidate partition single-peaked consistent *if the set of candidates C can be partitioned into at most k disjoint sets $C_1, \ldots, C_k$ with $C_1 \cup \ldots \cup C_k = C$ such that the profiles $\mathscr{P}[C_1], \ldots, \mathscr{P}[C_k]$ are single-peaked consistent.*

Let $CP(\mathscr{P})$ denote the smallest k such that $\mathscr{P}$ is k-candidate partition single-peaked consistent. Note that $CP(\mathscr{P}) \leq \left\lceil \frac{|C|}{2} \right\rceil$ always holds.

Decision Problems

We now introduce the seven problems we will study. We define the following problem for X $\in$ {Maverick, Candidate Deletion, Local Candidate Deletion, Additional Axes, Global Swaps, Local Swaps, Candidate Partition}.

X SINGLE-PEAKED CONSISTENCY	
Given:	An election $E = (C, V, \mathscr{P})$ and a positive integer k.
Question:	Is $\mathscr{P}$ k-X single-peaked consistent?

4 Results

4.1 Basic Results about Single-Peaked Profiles

We start with a simple observation which we will use in the proof of Theorem 4.6.

Lemma 4.1. *Let $\mathscr{P}$ be a preference profile containing the preference relation $\succ_1$: $c_1 \ldots c_n$ and its reverse $\succ_2$: $c_n \ldots c_1$. Then $\mathscr{P}$ is either single-peaked with respect to the axis $c_1 < \cdots < c_n$ (and its reverse) or it is not single-peaked at all.*

Proof. Since the vote $\succ_1$ ranks c_n last while the vote $\succ_2$ ranks c_1 last, these candidates have to be at the left-most and right-most position on any compatible axis. Note that c_1 is the peak in $\succ_1$. Hence this already determines the position of all other candidates. Consequently only two axes are possible: $c_1 < \cdots < c_n$ and $c_n < \cdots < c_1$. Since any preference profile is single-peaked with respect to $c_1 < \cdots < c_n$ if and only if it is single-peaked with respect to $c_n < \ldots < c_1$, we can focus without loss of generality on the former. ❑

Lemma 4.2 provides an alternative characterization of single-peaked consistency.

Lemma 4.2. *Given an election $(C, V, \mathscr{P})$, the profile $\mathscr{P}$ is not single-peaked consistent if and only if for all axes A there is some voter $v \in V$ and three candidates $c_i, c_j, c_k \in C$ such that $c_i > c_j > c_k$ on axis A, and $c_i \succ_v c_j$ holds as well as $c_k \succ_v c_j$.*

The following observation says that any subelection, i.e., an election with the same voters over a subset of the candidate set, of a single-peaked election is also single-peaked.

Lemma 4.3. *Let $(C, V, \mathscr{P})$ be a given election and $C' \subseteq C$. If $\mathscr{P}$ is single-peaked consistent then also $\mathscr{P}[C']$ is single-peaked consistent.*

In the constructions in our main results we will have to cascade two or more preference profiles. The following definition captures this notion.

Definition 4.4. *Let $(C_1, V, \mathscr{P}_1)$ and $(C_2, V, \mathscr{P}_2)$ be two elections with $C_1 \cap C_2 = \emptyset$. Furthermore, let $\mathscr{P}_1 = (\succ'_1, \ldots, \succ'_n)$ and $\mathscr{P}_2 = (\succ''_1, \ldots, \succ''_n)$. We define $\mathscr{P}_1 \circledcirc \mathscr{P}_2 = (\succ_1, \ldots, \succ_n)$, where for any $1 \le i \le n$ the linear order $\succ_i$ is defined by*

$$c \succ_i c' \text{ iff } (c, c' \in C_1 \text{ and } c \succ'_i c') \text{ or } (c, c' \in C_2 \text{ and } c \succ''_i c') \text{ or } (c \in C_1 \text{ and } c' \in C_2).$$

Note that $\mathscr{P}_1 \circledcirc \mathscr{P}_2$ is always a preference profile over $C_1 \cup C_2$.

Lemma 4.5. *Let $(C_1, V, \mathscr{P}_1)$ and $(C_2, V, \mathscr{P}_2)$ be two elections with $C_1 \cap C_2 = \emptyset$. Assume that*

- *$\mathscr{P}_1$ and $\mathscr{P}_2$ are single-peaked consistent with respect to the axes A_1 and A_2, respectively.*
- *The preference relations in $\mathscr{P}_2$ have at most 2 peaks.*
- *These (two) peaks are adjacent on the axis A_2.*

Then $\mathscr{P}_1 \circledcirc \mathscr{P}_2$ is single-peaked.

Proof. We are going to construct an axis A in a way that $\mathscr{P}_1 \circledcirc \mathscr{P}_2$ is single-peaked with respect to A. First we split A_2 in two parts A'_2 and A''_2. If $\mathscr{P}_2$ contains two peaks (which have to be adjacent), we split A_2 in between these two peaks. If $\mathscr{P}_2$ contains only one peaks, we split A_2 left of the peak (this is arbitrary). The new axis A is A'_2 followed by A_1 and then A''_2, i.e., $A'_2 > A_1 > A''_2$. The correctness proof of this construction is straight-forward. ❑

4.2 Relations between Notions of Nearly Single-Peakedness

Theorem 4.6 shows several inequalities that hold for the distance measures under consideration. We hereby show how these measures relate to each other. Notice that these inequalities do not have an immediate impact for a classical complexity analysis such as in Section 4.3.

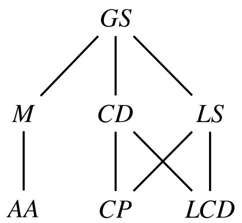
Theorem 4.6. *Let $\mathscr{P}$ be a preference profile. Then the following inequalities hold:*

(1) $LS(\mathscr{P}) \leq GS(\mathscr{P})$. *(4)* $LCD(\mathscr{P}) \leq LS(\mathscr{P})$. *(7)* $CP(\mathscr{P}) \leq CD(\mathscr{P}) + 1$.
(2) $LCD(\mathscr{P}) \leq CD(\mathscr{P})$. *(5)* $M(\mathscr{P}) \leq GS(\mathscr{P})$. *(8)* $CP(\mathscr{P}) \leq LS(\mathscr{P}) + 1$.
(3) $CD(\mathscr{P}) \leq GS(\mathscr{P})$. *(6)* $AA(\mathscr{P}) \leq M(\mathscr{P})$.

This list is complete in the following sense: Inequalities that are not listed here and that do not follow from transitivity do not hold in general. The resulting partial order with respect to $\leq$ is displayed in Figure 1 as a Hasse diagram.

Proof. Inequalities 1 and 2 are immediate consequences from the definitions since k-LS allows more swaps than k-GS and k-LCD allows more candidate deletions than k-CD. Inequalities 3 and 4 are due to the fact that swapping two candidates in a vote is at most as effective as removing one of these candidates. Similarly, for Inequality 5 observe that removing the corresponding voter is at least as effective as swapping two candidates in the vote. Concerning Inequality 6 observe that instead of deleting a voter we can always add an additional axis for this voter. Inequality 7 follows from the fact that putting each deleted candidate in its own partition leads to single-peakedness if deleting these candidates does.

In order to show Inequality 8 let $\mathscr{P}$ be k-local swaps single-peaked consistent. This means that there exists an axis A such that after performing at most k swaps per voter, $\mathscr{P}$ becomes single-peaked with respect to A. Without loss of generality assume that the axis A is $c_1 < c_2 < \ldots < c_n$. We now partition the candidates in $k+1$ sets $S_0, \ldots, S_k$. This is done by putting the i-th smallest element of A into the (i modulo $k+1$)-th set. Since we assume that A is $c_1 < c_2 < \ldots < c_n$, we can equivalently say that c_i is put into the (i modulo $k+1$)-th set, i.e., the c_1 in S_1, the c_2 in S_2, the c_k in S_k and c_{k+1} in S_0. Let $S \in \{S_0, \ldots, S_k\}$. Towards a contradiction assume that $\mathscr{P}[S]$ is not single-peaked with respect to $A[S]$. By Lemma 4.2 there exists some voter $v \in V$ and three candidates $c_i, c_j, c_k \in C$ such that $c_i < c_j < c_k$ on axis $A[S]$ (or equivalently $i < j < k$), $c_i \succ_v c_j$ and $c_k \succ_v c_j$. On axis A the distance between c_i and c_j respectively c_j and c_k is at least $k+1$, i.e., at least k elements lie in between them. We know that at most k swaps in $\succ_v$ can make this profile single-peaked with respect to A. Let $\succ'_v$ denote this swapped vote. Necessarily these swaps have to either cause that $c_j \succ'_v c_{j-1} \succ'_v \ldots \succ'_v c_{i+1} \succ'_v c_i$ holds or that $c_j \succ'_v c_{j+1} \succ'_v \ldots \succ'_v c_{k-1} \succ'_v c_k$ holds in $\succ'_v$ (depending whether the peak of $\succ'_v$ is right or left of c_j). Let us focus on the case that the swaps ensure that $c_j \succ'_v c_{j-1} \succ'_v \ldots \succ'_v c_{i+1} \succ'_v c_i$ – the other case is analogous. For $\succ_v$, contrary to $\succ'_v$, it holds that $c_i \succ_v c_j$. Hence these swaps have to cause that $c_j \succ'_v c_i$ holds. In addition, at least k elements, namely $c_{i+1}, \ldots, c_{j-1}$, have to be in between them. This requires at least $k+1$ swaps which contradicts the



GS ... Global Swaps
M ... Maverick
CD ... Candidate Deletion
LS ... Local Swaps
AA ... Additional Axes
CP ... Candidate Partition
LCD ... Local Candidate Deletion

Figure 1: Hasse diagram of the partial order described in Theorem 4.6.

fact that at most k swaps suffice. Therefore for all partition sets S, $\mathscr{P}[S]$ is single-peaked consistent and $CP(\mathscr{P}) \leq LS(\mathscr{P}) + 1$.

It remains to show that these are indeed all inequalities. This can be done by providing counterexamples for each remaining case. ❑

4.3 Complexity of Nearly Single-Peaked Consistency

Let us first introduce a lemma which we will use in the proofs of the theorems below.

Lemma 4.7. *We are given a set of candidates $C = \{a,b,c,d\}$ and three preference relations $\succ_v$, $\succ_e$ and $\succ_{ne}$, where the candidates are ranked as follows:*

- $a \succ_v c \succ_v b \succ_v d$,
- $c \succ_e b \succ_e d \succ_e a$ *and*
- $d \succ_{ne} c \succ_{ne} b \succ_{ne} a$.

Then the preference profile $(\succ_v, \succ_e)$ is single-peaked with respect to the axis $a > c > b > d$ and $(\succ_e, \succ_{ne})$ is single-peaked with respect to the axis $d > c > b > a$. The profile $(\succ_v, \succ_{ne})$ is not single-peaked consistent.

We start with maverick single-peaked consistency where we show NP-hardness via a reduction from the clique problem, one of the standard NP-complete problems (see, e.g., [GJ79]).

Theorem 4.8. MAVERICK SINGLE-PEAKED CONSISTENCY *is* NP*-complete.*

Proof. To show hardness we reduce from CLIQUE. Let $G = (V_G, E_G)$ be the graph in which we look for a clique of size s. Furthermore, let $V_G = \{v_1, \ldots, v_n\}$ be the set of vertices and E_G the set of edges. Each vertex v_i has four corresponding candidates $c_i^1, \ldots, c_i^4$. We consequently have $C = \{c_1^1, \ldots, c_1^4, c_2^1, \ldots, c_2^4, \ldots, c_n^1, \ldots, c_n^4\}$. The voters directly correspond to vertices. Therefore we define, by slight abuse of notation, $V = \{v_1, \ldots, v_n\}$.

In order to define the preference relations we introduce three functions creating partial votes. In the following definition let $a,b,c,d \in C$.

$$\begin{aligned} f_v(a,b,c,d) &= a \succ c \succ b \succ d \\ f_e(a,b,c,d) &= c \succ b \succ d \succ a \\ f_{ne}(a,b,c,d) &= d \succ c \succ b \succ a \end{aligned}$$

If we consider f_v, f_e and f_{ne} as preference relations then observe that by Lemma 4.7 (f_v, f_e) and (f_e, f_{ne}) are single-peaked consistent but (f_v, f_{ne}) is not.

Next we define a mapping $p(i,j)$ to a total order over the candidates $\{c_j^1, \ldots, c_j^4\}$.

$$p(i,j) = \begin{cases} f_v(c_j^1, c_j^2, c_j^3, c_j^4) & \text{if } i = j \\ f_e(c_j^1, c_j^2, c_j^3, c_j^4) & \text{if } \{i,j\} \in E_G \\ f_{ne}(c_j^1, c_j^2, c_j^3, c_j^4) & \text{if } \{i,j\} \notin E_G \end{cases}$$

The intuition behind function $p(i,j)$ is to encode a row of the adjacency matrix of G as a vote in the preference profile $\mathscr{P}$. To this end, we put in "cell" (i,j) the result of f_e if there is an edge between i and j. In case there is no edge between i and j we put the result of f_{ne} in cell (i,j). In the special case $i = j$ (we are in the diagonal of the matrix) we put the result of f_v in the cell.

Let the partial profiles representing the columns of the adjacency matrix be defined as $\mathscr{P}_j = (p(1,j),\dots,p(n,j))$, for $1 \le j \le n$. We are now going to define the preference profile $\mathscr{P} = (\succ_1,\dots,\succ_n)$ by

$$\mathscr{P} = \mathscr{P}_1 \otimes \mathscr{P}_2 \otimes \dots \otimes \mathscr{P}_n.$$

To conclude the construction let $E = (C,V,\mathscr{P})$ and $k = n-s$, i.e., we are allowed to delete k mavericks from E in order to obtain a single-peaked profile. The intention behind the construction is that the voters in a single-peaked profile will correspond to a clique. We claim that G has a clique of cardinality s if and only if it is possible to remove k voters from $\mathscr{P}$ in order to make the resulting preference profile single-peaked consistent.

"$\Rightarrow$" Assume that there is a clique $I = \{v_{i_1},\dots,v_{i_s}\}$ with $|I| = s$. Let $\mathscr{P}' = (\succ_{i_1},\dots,\succ_{i_s})$. By that we keep only those voters whose corresponding vertices are contained in the clique I. Observe that the election $E' = (C,I,\mathscr{P}')$ can be obtained by deleting $k = n-s$ mavericks from the election E, $|V \setminus I| = k$. It remains to show that E' is indeed single-peaked consistent. Remember that we denoted the preference relations in the j-th "column" of the profile by $\mathscr{P}_j$. By $\mathscr{P}'_j$ we denote the j-th "column" of a profile considering only the voters from $\mathscr{P}'$. Since I is a clique, for each $x,y \in I$, $x \ne y$, there is an edge $\{x,y\} \in E_G$. Thus the profile cannot contain an instantiation of f_v and of f_{ne} in the same column. By Lemma 4.7, all profiles $\mathscr{P}_j$ with $1 \le j \le n$ are single-peaked consistent. In order to be able to apply Lemma 4.5, all conditions have to be checked. First, notice that the profiles $\mathscr{P}'_j$ and $\mathscr{P}'_{j'}$, for $1 \le j < j' \le n$, do not share any candidates and are single-peaked consistent. Furthermore, each of the profiles has at most two peaks. Each column contains either instantiations of f_v and f_e or instantiations of f_e and f_{ne}. Otherwise it would not be single-peaked consistent. But then there are only two top-ranked candidates, i.e., either the candidates top-ranked by f_v and f_e, or the candidates top-ranked by f_e and f_{ne}. Finally, the two top-ranked candidates of $\mathscr{P}'_j$ have to be adjacent on the axis which gives single-peaked consistency. Consider again Lemma 4.7. For (f_v, f_e) the top-ranked candidates a and c are adjacent on the axis $a > c > b > d$. The same holds for (f_e, f_{ne}) with axis $d > c > b > a$ and c, d as top-ranked candidates. Since all conditions are fulfilled, we can iteratively apply Lemma 4.5. Therefore, $\mathscr{P}'_1 \otimes \mathscr{P}'_2, (\mathscr{P}'_1 \otimes \mathscr{P}'_2) \otimes \mathscr{P}'_3, \dots, (\mathscr{P}'_1 \otimes \dots) \otimes \mathscr{P}'_n$ and hence also $\mathscr{P}'$ are single-peaked consistent.

"$\Leftarrow$" Assume that $E' = (C,V',\mathscr{P}')$ is an election that has been obtained from E by deleting k voters such that $\mathscr{P}'$ is single-peaked. Consequently $|V'| = s$. Let $V' = \{v_{i_1},\dots,v_{i_s}\}$ and $\mathscr{P}' = (\succ_{i_1},\dots,\succ_{i_s})$.

We claim that V' is a clique in G. By Lemma 4.3 we know that each of the n columns $(\mathscr{P}'_1,\dots,\mathscr{P}'_n)$ of $\mathscr{P}'$ is single-peaked consistent. Then, by Lemma 4.7, each column must not contain an instance of f_v together with an instance of f_{ne}. (Otherwise the respective column would not be single-peaked consistent!) Observe that by construction each vote (in $\mathscr{P}'$) contains an instance of f_v in some column. But then each vertex must be adjacent to all other vertices – in other words V' is a clique. ❑

We now turn to additional axes single-peaked consistency. Here we make use of a similar construction as presented in Theorem 4.8 with the difference that we now show NP-hardness via a reduction from the partition into cliques problem, which is also one of the standard NP-complete problems (see, e.g., [GJ79]).

Theorem 4.9. ADDITIONAL AXES SINGLE-PEAKED CONSISTENCY *is* NP-*complete.*

Proof. Hardness is shown by a reduction from PARTITION INTO CLIQUES. For the reduction we use the same transformation as presented in the proof of Theorem 4.8 to obtain an election. Then we set $k = s-1$, i.e., we are searching for a partition of the voters into s disjoint sets such that each of the partitions is single-peaked consistent. Due to the one-to-one correspondence between voters and vertices we can use the partition of the vertices to obtain a partition of the voters and vice versa. With arguments similar to the proof of Theorem 4.8 one can show that a set of vertices is a clique if and only if the corresponding profile is single-peaked consistent. ❑

In the proofs of our last two results, we will provide reductions from the NP-complete problem MINIMUM RADIUS, which was shown to be NP-complete in [FL97] and is defined as follows:

MINIMUM RADIUS	
Given:	A set of strings $S \subseteq \{0,1\}^n$ and a positive integer s.
Question:	Has S a radius of at most s, i.e., is there a string $\alpha \in \{0,1\}^n$ such that each string in S has a Hamming distance of at most s to α?

Theorem 4.10. LOCAL CANDIDATE DELETION SINGLE-PEAKED CONSISTENCY *is* NP-*complete.*

Proof. A MINIMUM RADIUS instance is given by $S \subseteq \{0,1\}^n$, the set of binary strings, and a positive integer s. Given a string β, let $\beta(k)$ denote the bit value at the k-th position in β. We are going to construct an LCD SINGLE-PEAKED CONSISTENCY instance. Each string in $S = \{\beta_1, \ldots, \beta_m\}$ will correspond to a voter. Each bit of the strings corresponds to two candidates. In addition, we have $2 \cdot m \cdot s + 2$ extra candidates. Consequently, we have $C = \{c_1^1, c_1^2, c_2^1, c_2^2, \ldots, c_n^1, c_n^2, c'_1, \ldots, c'_{ms+1}, c''_1, \ldots, c''_{ms+1}\}$.

We define the preference profile with the help of two functions creating total orders.

$$f_0(a,b) = a \succ b \qquad\qquad f_1(a,b) = b \succ a$$

The vote $\succ_k$, for each $k \in \{1, \ldots, m\}$, is of the form

$$c'_1 \ \ldots\ c'_{ms+1}\ \ f_{\beta_k(1)}(c_1^1, c_1^2)\ \ f_{\beta_k(2)}(c_2^1, c_2^2)\ \ \ldots\ \ f_{\beta_k(n)}(c_n^1, c_n^2)\ \ c''_1\ \ldots\ c''_{ms+1}.$$

Furthermore, let $\succ_k^r$, $1 \leq k \leq m$, denote the reverse order of $\succ_k$. The preference profile $\mathscr{P}$ is now defined as $(\succ_1, \ldots, \succ_n, \succ_1^r, \ldots, \succ_n^r)$. We claim that $(V, C, \mathscr{P})$ is s-LCD single-peaked consistent if and only if S has a radius of at most s.

"$\Leftarrow$" Suppose that S has a radius of at most s, i.e., there is a string $\alpha \in \{0,1\}^n$ with Hamming distance at most s to each $\beta \in S$. We consider the following axis A:

$$c'_1 > \ldots > c'_{ms+1} > f_{\alpha(1)}(c_1^1, c_1^2) > f_{\alpha(2)}(c_2^1, c_2^2) > \ldots f_{\alpha(n)}(c_n^1, c_n^2) > c''_1 > \ldots > c''_{ms+1}.$$

We claim that $\mathscr{P}$ is single-peaked with respect to A after deleting at most s candidates in each vote. The deletions for vote $\succ_k$, $k \in \{1, \ldots, m\}$, are the following: We delete candidate c_i^1 in $\succ_k$ if and only if $\alpha(i) \neq \beta_k(i)$. The deletions in $\succ_k^r$ are exactly the same as in $\succ_k$. These are at most s deletions since the Hamming distance between α and every $\beta \in S$ is at most s. After these deletions all votes are either subsequences of A or its reverse. Hence we obtain a single-peaked consistent profile.

"$\Rightarrow$" Let $\mathscr{P}'$ be the partial, single-peaked consistent profile that was obtained by deleting at most s candidates in each vote. First, note that some $c' \in \{c'_1, \ldots, c'_{ms+1}\}$ has not been deleted in any vote since in total at most $m \cdot s$ many different candidates can be deleted. In the same way let $c'' \in \{c''_1, \ldots, c''_{ms+1}\}$ be a candidate that has not been deleted in any vote. Now let us consider the profile $\mathscr{P}'[\{c', c'', c_i^1, c_i^2\}]$ for any $i \in \{1, \ldots, n\}$. We claim that α, defined in the following way, has a Hamming distance of at most s to all bitstrings in S.

$$\alpha(k) = \begin{cases} 0 & \text{if } \mathscr{P}' \text{ contains the vote } c' \succ c_i^1 \succ c_i^2 \succ c'', \\ 1 & \text{if } \mathscr{P}' \text{ contains the vote } c' \succ c_i^2 \succ c_i^1 \succ c'', \\ 1 & \text{otherwise.} \end{cases}$$

First, observe that case 1 and 2 cannot occur at the same time since then $\mathscr{P}'$ would not be single-peaked consistent. This is because $\mathscr{P}'[\{c', c'', c_i^1, c_i^2\}]$ also contains the vote $c'' \succ \ldots \succ c'$, where the dots indicate that c_i^1 and c_i^2 might also appear in this vote (between c'' and c'). Furthermore, Let

$\beta_j \in S$ from some $j \in \{1, \dots, n\}$. Note that if at any position i, $\beta_j(i) \neq \alpha(i)$ then either c_i^1 or c_i^2 had to be deleted in the vote $\succ_j$. Hence the set $\{k \in \{1, \dots, m\} \mid \alpha(i) \neq \beta_j(i)\}$ cannot contain more than s elements because this would require more than s candidate deletions in the corresponding vote $\succ_j$. Hereby we have shown that the Hamming distance of α and β_j is at most s. ❑

Theorem 4.11. LOCAL SWAPS SINGLE-PEAKED CONSISTENCY *is* NP-*complete.*

Proof. We use the same construction as in the proof of Theorem 4.10. It holds that $(V, C, \mathscr{P})$ is s-LS single-peaked consistent if and only if S has a radius of at most s. This can be shown similarly to the proof of Theorem 4.10 except that we swap elements instead of deleting them. ❑

5 Conclusions and Open Questions

We have investigated the problem of nearly single-peaked consistency. To this end, we have formally defined two notions of nearly single-peakedness suggested by Escoffier, Lang, and Öztürk [ELÖ08]. Furthermore, we have introduced three new notions of nearly single-peakedness. We have drawn a complete picture of the relations between all the notions of nearly single-peakedness discussed in this paper. For four notions we have shown that deciding single-peaked consistency is NP-complete. An obvious direction for future work is to pinpoint the complexity of the remaining three problems. It is noteworthy in this regard that a distance measure has been studied very recently which admits a polynomial time algorithm for nearly single-peaked consistency [EFS12].

NP-completeness, however, does not rule out the possibility of algorithms that perform well in practice. One approach is to search for fixed-parameter algorithms. For example, it might be that k-maverick single-peaked consistency can be decided by a fixed-parameter algorithm, i.e., an algorithm with runtime $f(k) \cdot poly(n)$ for some computable function f. A second approach is the development of approximation algorithms since nearly single-peaked consistency can also be seen as an optimization problem.

Another interesting direction for future work is extending our models to manipulative behavior, such as manipulation, control, and bribery. That is, assuming we have a nearly single-peaked electorate according to one of our notions, how hard is a manipulative action under a certain voting rule computationally? This line of work has already been started in [FHH11] for some distance measures. Finally, there might be further useful and natural distance measures regarding single-peakedness to be found.

Acknowledgments: We thank the anonymous reviewers for their helpful comments.

References

[BBHH10] F. Brandt, M. Brill, E. Hemaspaandra, and L. A. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 715–722. AAAI Press, 2010.

[BEH+10] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. In J. Laslier and R. Sanver, editors, *Handbook on Approval Voting*, chapter 10, pages 199–251. Springer, 2010.

[Bla48] D. Black. On the rationale of group decision making. *Journal of Political Economy*, 56(1):23–34, 1948.

[BT86] J. Bartholdi and M. Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4):165 – 169, 1986.

[BTT89] J. Bartholdi, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[BTT92] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[Con09] V. Conitzer. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, 35:161–191, 2009.

[DKNS01] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference*, pages 613–622. ACM Press, 2001.

[EFS09] E. Elkind, P. Faliszewski, and A. M. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT-2009)*, pages 299–310, 2009.

[EFS12] E. Elkind, P. Faliszewski, and A. M. Slinko. Clone structures in voters' preferences. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC-12)*, pages 496–513. ACM, 2012.

[ELÖ08] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 366–370. IOS Press, 2008.

[ER97] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.

[FHH09] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[FHH11] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of manipulative attacks in nearly single-peaked electorates. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2011)*, pages 228–237, 2011.

[FHHR11] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, 209(2):89–107, 2011.

[FL97] M. Frances and A. Litman. On covering problems of codes. *Theory of Computing Systems*, 30:113–119, 1997.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[GMHS99] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of recommender systems. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 434–435. ACM Press, 1999.

[MN08] T. Meskanen and H. Nurmi. Closeness counts in social choice. In M. Braham and F. Steffen, editors, *Power, Freedom, and Voting*, pages 289–306. Springer, 2008.

[Wal07] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07)*, pages 3–8. AAAI Press, 2007.

Gábor Erdélyi
Institut für Betriebswirtschaftslehre
Universität Siegen
Siegen, Germany
Email: `erdelyi@wiwi.uni-siegen.de`

Martin Lackner
Vienna University of Technology
Vienna, Austria
Email: `lackner@dbai.tuwien.ac.at`

Andreas Pfandler
Vienna University of Technology
Vienna, Austria
Email: `pfandler@dbai.tuwien.ac.at`

Models of Manipulation on Aggregation of Binary Evaluations

Elad Dokow and Dvir Falik

Abstract

We study a general aggregation problem in which a society has to determine its position on each of several issues, based on the positions of the members of the society on those issues. There is a prescribed set of feasible evaluations, i.e., permissible combinations of positions on the issues. Among other things, this framework admits the modeling of preference aggregation, judgment aggregation, classification, clustering and facility location. An important notion in aggregation of evaluations is strategy-proofness. In the general framework we discuss here, several definitions of strategy-proofness may be considered. We present here 3 natural *general* definitions of strategy-proofness and analyze the possibility of designing an anonymous, strategy-proof aggregation rule under these definitions.

1 Introduction

There is, by now, a significant body of literature on the problem of aggregating binary evaluations. A society has to determine its position (yes/no) on each of several issues, based on the positions of the members of the society on those issues. There is prescribed set X of feasible evaluations, i.e., permissible combinations of positions on the issues (X may be viewed as a subset of $\{0,1\}^m$, where m is the number of issues). The members of the society report their opinion to an aggregation mechanism, called the *aggregator*, which outputs society's aggregated opinion. Many examples include preference aggregation (where the issues are pairwise comparisons and feasibility reflects rationality), and judgment aggregation (where the issues are logical propositions and feasibility reflects consistency) can be presented by this framework. We shall refer to this framework as Judgment Aggregation throughout the paper, as this model is actually as general as the entire framework.

This paper deals with introducing a general definition of manipulations and strategy-proofness to this model. Generally speaking, we assume that each member of the society has some preference over the possible outcomes, which is derived from her true opinion on the issues. Under this assumption, it may not always be the rational course of action for a member of the society to report her true opinion to the aggregator. Such an occurrence is called a *manipulation* of the aggregator. An aggregator which is immune to manipulations is called *strategy-proof*. There is no canonical way to define the concept of manipulation in judgment aggregation, and at least one choice of definition has been studied. In this paper we wish to initiate a systematic study of the range of general definitions of manipulations for Judgment aggregation and analyze the possibility of designing strategy-proof aggregators for given evaluation spaces under a given definition of manipulation.

We present a few applications-examples taken from several distinct areas, that can all be modeled via the judgment aggregation framework:

Preference Aggregation: In this setting, the society wishes to rank k alternatives, in order of preference, where each voter has its own private order of preference. This problem has been studied since the days of the French revolution, by the Marquis de Condorcet. We will only address in this paper the case where the ranking has to be full - i.e. there is always a strict preference between 2 alternatives[1]. The set of issues is the set of pairwise

[1]There are works that deal with the more general framework, where the preferences are not strict, see,

preferences between every 2 alternatives, so $m = \binom{k}{2}$. Each pair of alternatives may have 2 possible preferences, so the full opinion has a binary encoding. The permissible evaluations are the preferences that encode a full transitive order.

For example, when $k = 3$, the set of alternatives is $\{a, b, c\}$, the set of issues is $\{a \succ b, b \succ c, c \succ a\}$, and the permissible evaluations are all possible evaluations except 000 and 111, which encode a non-transitive order.

Condorcet noticed that a specific natural aggregator, that chooses in each issue the majority opinion of the society in that issue, does not always produce permissible evaluations. This is known as "Condorcet's Paradox". Condorcet's Paradox motivated the study of social choice theory, beginning in Arrow's theorem [Arr63].[2]

Judgment Aggregation: In the last decade, there is a growing body of work in the field of judgment Aggregation, where judges need to come to a decision on a set J of connected issues. The connection between the issues is expressed by a set of permissible evaluations $X \subseteq \{0,1\}^J$. The canonical example in this context is the *doctrinal paradox* (also called *the discursive dilemma*), in which a court has to decide whether a defendant is guilty. In order to declare him guilty, they must hold the opinion that he has committed the crime and that he was sane at the time. The set of permissible evaluation, therefore, is

$$X = \{(p, q, r) | r = p \wedge q\}$$

The so called "paradox" arises when a majority of the judges think that the defendant has committed the crime, and a majority of the judges believe he was sane at the time, but only a minority of the judges believe both to hold.

	$a \succ b$	$b \succ c$	$c \succ a$
Voter 1	1	1	0
Voter 2	0	1	1
Voter 3	1	0	1
Aggr.	1	1	1

Table 1: Condorcet's paradox

	Murdered	Sane	Guilty
Judge 1	0	1	0
Judge 2	1	0	0
Judge 3	1	1	1
Majority	1	1	0

Table 2: Doctrinal Paradox

Many works done in recent years discussed this general framework (See the survey [LP10]). In particular, the conditions on X for which Arrow's theorem holds has been extensibly studied.

Classification A set of m points has to be classified, and there is a prescribed set of classifiers. For instance, consider the case where the points lie in $\mathbb{R}^k$, and the classifiers are all the linear half-spaces. The society is composed of n agents, each has its own classification of the points, and the aggregator must select a classifier based on the opinions of the agents.

This problem fits into our framework when the classifiers are encoded as the vector of their classification of all the points.

For example, consider the points to be $\{(0,0), (0,1), (1,0), (1,1)\}$. The possible linear classifiers in this case are all classifiers except for 0110 and 1001: $X = \{0,1\}^4 \setminus \{0110, 1001\}$

Facility Location:

e.g. [Arr63].

[2] Arrow's theorem states that it is impossible to design a social aggregator that satisfies some natural conditions. It is natural to assume that for every social aggregator, the aggregated order is always a transitive order (consistent), that it agrees with a unanimous vote (Pareto optimal), and that it is influenced by the opinions of more than one voter (non-dictatorial). Condorcet's proposed aggregator satisfied the property that its decision on the social preferences between alternatives a and b depends only on the individual preferences between a and b. This property is known as Independence of Irrelevant alternatives, or IIA. Arrow showed that a social aggregator on 3 or more alternatives that satisfies IIA, cannot be consistent, Pareto-optimal and non-dictatorial.

In this problem[DFMN12], an aggregator is given k points in some metric space, and is required to choose a location for a facility that services all these points. The location of each point is reported to the aggregator by a single agent, which may or may not be truthful. The aggregator should optimize the distance of the chosen location from the locations of the points.

We can encode this problem under our framework in the case that the metric space that is used is isomorphic to an induced subgraph of the Boolean hypercube equipped with the Hamming metric. The set X of permissible evaluations will be the set of vertices in the Boolean hypercube corresponding to the given metric space. For instance, a simple cycle on $2m$ vertices can be encoded as $X = \left\{1^i 0^{m-i} | i \in \{0..m\}\right\} \cup \left\{0^{m-i} 1^i | i \in \{1..m-1\}\right\}$

1.1 Strategy-proofness and manipulations

A variant of preference aggregation is *social choice*, where the social aggregator is required to choose society's preferred alternative, based on the voters' preferences. Gibbard and Satterthwaite [Gib73, Sat75] showed an impossibility theorem for social choice aggregators. Their theorem deals with the game-theoretic notion of *manipulations*. A manipulation is a situation where a voter can mis-report her preference and obtain a preferable alternative, accoridng to her true preference. An aggregator is called *strategy-proof* if it allows no manipulations. The theorem states that there is no non-dictatorial social aggregator that is strategy-proof (for at least 3 alternatives).

This work aims at generalizing the concept of manipulation to the general setting of judgment aggregation. However, in this context, the preference of a voter over all possible results is not clear from her opinion, and each problem can have a different interpretation of this notion. [3]

In order to reach a general definition of manipulation, we assume that each voter desires the aggregated evaluation to agree with her personal evaluation in all or some of the issues. Since there may be situations where some of the issues change for the better and some for the worse (in the manipulator's view), there is still a degree of freedom in the choice of a definition of a manipulation. This work discusses 3 natural definitions of the concept of manipulation on judgment aggregation. One of the definitions was defined and discussed in [NP10, DL07], and it leads to impossibility results similar to those that were mentioned here. The other two definitions allow non-dictatorial aggregators, and we will discuss the construction of such aggregators in the general case.

Consider an aggregator for preference aggregation on 3 alternatives, that uses the *plurality* method. It selects the ranking that was voted for the highest number of times. In case of a tie, it uses a lexicographical order to choose the ranking. Consider the following profile:

	$a \succ b$	$b \succ c$	$c \succ a$
Voter 1	1	1	0
Voter 2	0	1	1
Voter 3	1	0	1
Aggr.	1	1	0

	$a \succ b$	$b \succ c$	$c \succ a$
Voter 1	1	1	0
Voter 2	1	0	1
Voter 3	1	0	1
Aggr.	1	0	1

In the profile to the left, the second voter has the society agreeing with her on the second issue - $b \succ c$, and disagreeing with her in the other issues. She can change this when reporting a different opinion. In the profile to the right, society agrees with her original opinion in the third issue - $c \succ a$, and disagrees with her on the other issues.

[3] Note that there are works that extend the setting of GS to multi-issue voting, e.g. [?]. This is not the setting we analyze.

If her main interest was in getting society to agree with her in the third issue, then she has successfully manipulated the aggregator.

If her main interest was to get the society to agree with her in as many issues as possible, then she has not manipulated the aggregator, as in both cases the aggregated opinion agreed with her in only 1 issue.

The following table shows a different scenario:

	$a \succ b$	$b \succ c$	$c \succ a$
Voter 1	1	0	1
Voter 2	0	1	1
Voter 3	0	1	0
Aggr.	1	0	1

	$a \succ b$	$b \succ c$	$c \succ a$
Voter 1	1	0	1
Voter 2	0	1	0
Voter 3	0	1	0
Aggr.	0	1	0

In the profile to the left, the second voter has the society agreeing with her on the third issue - $c \succ a$, and disagreeing with her in the other issues. When she reports a different opinion, as shown in the profile to the right, society agrees with her original opinion on the first 2 issues. She has gained in the number of issues the society agrees with her. However, she has lost the agreement with the society on the third issue.

The following table is third and final scenario:

	$a \succ b$	$b \succ c$	$c \succ a$
Voter 1	1	0	1
Voter 2	0	1	1
Voter 3	0	0	1
Aggr.	1	0	1

	$a \succ b$	$b \succ c$	$c \succ a$
Voter 1	1	0	1
Voter 2	0	0	1
Voter 3	0	0	1
Aggr.	0	0	1

In the profile to the left, the second voter has the society agreeing with her on the third issue - $c \succ a$, and disagreeing with her in the other issues. When she reports a different opinion, as shown in the profile to the right, society agrees with her original opinion on the first and last issues. She has gained agreement in the first issue and did not lose agreement on any of the other issues.

When designing an aggregator, we need to know what type of manipulations we wish to be immune against. A maximal requirement is to be immune from manipulations that gain in any of the issues (we will call these *partial manipulations*). A minimal requirement is immunity from manipulations that don't lose agreement in any of the issues (we will call these *full manipulations*). There could be other types of manipulations in between, for instance, manipulations that gain in the number of issues agreed with the society (we will call these *Hamming manipulations*).

1.2 Structure of the paper and results

In section 2 we present the formal model used throughout the paper. We then dedicate a chapter for each of the 3 types of manipulations mentioned above. Section 3 discusses the *partial manipulation*, section 4 discusses the *full manipulation*, and section 5 discusses the *Hamming manipulation.*

Partial manipulation: Partial manipulation was already discussed in previous works. We state known results here for completion. These results characterize evaluation spaces X for which the only partial manipulation free aggregators are dictatorial. These results are based on the connection between partial manipulations and IIA.

Full manipulation We show that there is a family of non-dictatorial full manipulation free aggregators for every evaluation space X. In addition, for every evaluation space X

some members of this family are also Hamming manipulation free. We next turn to the question of anonymous full manipulation free aggregators. For every evaluation space X, we construct a family of aggregators that are anonymous and full manipulation free. These aggregators are also "close" to being partial manipulation free in some sense. We also show that when the welfare of a voter is defined as the Hamming distance between its opinion and society's decision, the social welfare maximizer is a full manipulation free aggregator.

Hamming manipulation Again, we discuss the possibility of constructing a *anonymous* Hamming manipulation free aggregator. Since every *Hamming manipulation free* aggregator is also a *full manipulation free* aggregator, we try and characterize the evaluation spaces X for which the full manipulation free anonymous aggregators mentioned above are also Hamming manipulation free. We do not have a full characterization of these aggregators. We describe some conditions that affect the Hamming strategy proofness of these aggregators, based on the geometry of the evaluation space. We apply these techniques to demonstrate that in the case of preference aggregation on 3 alternatives these aggregators are Hamming manipulation free, and for 4 alternatives we show that a subfamily of these aggregators are not Hamming manipulation free.

2 The setting

We consider a finite, non-empty set of issues J. For convenience, if there are m issues in J, we identify J with the set $\{1, ...m\}$ of coordinates of vectors of length m. A vector $x = (x_1, ..., x_m) \in \{0,1\}^m$ is an *evaluation.* We assume that some non-empty subset X of $\{0,1\}^m$ is given. The evaluations in X are called feasible, the others are infeasible. We shall also use this terminology for partial evaluations: for a subset of issues K, a K-evaluation is feasible if it lies in the projection of X on the coordinates in K, and is infeasible otherwise. A *society* is a finite, non-empty set N. For convenience, if there are n individuals in N, we identify N with the set {1,...,n}. If we specify a feasible evaluation $x^i = (x^i_1, ..., x^i_m) \in X$ for each individual $i \in N$, we obtain a profile of feasible evaluations $\mathbf{x} = (x^i_j) \in X^n$. We may view a profile as an $n \times m$ matrix all of whose rows lie in X. We use superscripts to indicate individuals (rows) and subscripts to indicate issues (columns). An *aggregator* for N over X is a mapping $f : X^n \to X$. It assigns to every possible profile of individual feasible evaluations, a social evaluation which is also feasible. Any aggregator f may be written in the form $f = (f_1, ..., f_m)$ where f_j is the j-th component of f. That is, $f_j : X^n \to \{0,1\}$ assigns to every profile the social position on the j-th issue. We write $\mathbf{x} = (x^i, x^{-i})$ to distinguish between the opinion of the i-th individual and the opinions of the rest of the society.

Definition 2.1: Independence of Irrelevant Alternatives (IIA) An aggregator is called IIA if the society's position on any given issue depends only on the individual positions on that same issue. $\forall \mathbf{x}, \mathbf{y} \in X^n, j \in J, (\mathbf{x}_j = \mathbf{y}_j) \Rightarrow (f_j(\mathbf{x}) = f_j(\mathbf{y}))$

Definition 2.2: Anonymity An aggregator is called *anonymous* if it does not depend on the order of the evaluations in the profile, i.e. for every permutation p of the evaluators, $f(x^{p(1)}, ...x^{p(n)}) = f(x^1, ...x^n)$.

Definition 2.3: Monotonicity An aggregator is called *monotone* if for every issue $j \in J$, changing an individual's position on j never results in a change of society's position on j in the opposite direction.

Definition 2.4: Dictatorship An aggregator is called *dictatorial* if it obeys the opinion of only one of the evaluators: There exists an evaluator $i \in N$ such that $f(\mathbf{x}) = x^i$.

2.1 Strategic Voting and Strategy-Proofness

We now assume that each voter desires the social evaluation to agree with her personal evaluation in all or some of the issues. Under this assumption, it may not always be the rational choice for the voter to declare her true evaluation to the aggregator. Given that the other voters voted in a specific way, lying about her evaluation may change society's position on certain issues to match hers, making society's position "closer" to hers, under some definition of closeness. An evaluator i is said to have a manipulation of an aggregator f in a profile $\mathbf{x} \in X^n$ if she can report a false evaluation y in a way that $w = f(y, x^{-i})$ is preferred by her over $z = f(x^i, x^{-i})$. y is called a manipulation of i over $\mathbf{x}$.

What is left to decide is, when w is preferred over z, according to x^i. For an issue $j \in J$, if $w_j = x^i_j \neq z_j$, we shall call w *j-preferable* over z according to x^i. If $w_j = z_j$, then we say that w and z are *j-indifferent* to each other according to x^i. It is natural to assume that under any definition of manipulation, if y is a manipulation of i over $\mathbf{x}$, then there must be at least one issue $j \in J$ such that w is j preferable over z according to x^i. It is also natural to assume that under any definition of manipulation, if for every $j \in J$, z is j-indifferent to w or j-preferable over w according to x^i, then y is not a manipulation of i over $\mathbf{x}$.

We will base our definitions of manipulation on these assumptions.

Definition 2.5: Partial Manipulation: If there exists an issue $j \in J$ such that w is j preferable over z according to x^i, then y is a *partial manipulation* of i over $\mathbf{x}$.

Definition 2.6: Full Manipulation: If there exists an issue $j \in J$ such that w is j preferable over z according to x^i, and for every issue $j' \in J$, w is j'-preferred over z or j'-indifferent to z according to x^i, then y is a *full manipulation* of i over $\mathbf{x}$.

All possible definitions of manipulations that fit our assumptions lie between these two definitions. A natural and interesting choice of a definition of manipulation is based on the (Weighted) Hamming metric. For two vectors $x, y \in \{0,1\}^m$, we define their weighted Hamming distance with weight $\omega \in \mathbb{R}^m_+$ where $\sum_{j=1}^m \omega_j = 1$ as

$$d_w(x, y) = \sum_{j=1}^{m} \omega_j |x_j - w_j|$$

.

We will deal with the case when all the voters share the same weight function ω of the issues, and use the following definition:

Definition 2.7: Hamming Manipulation: If $d_\omega(x^i, w) < d_\omega(x^i, z)$, then y is a ω-*Hamming manipulation* of i over $\mathbf{x}$.

If ω is uniform over the issues, we omit it from the notation[4].

In subsection 1.1, the first example was a partial manipulation which was not a Hamming nor full manipulation, and the second example was a partial manipulation and a Hamming manipulation, but not a full manipulation. The third example is of a full manipulation. A manipulation of any type is also a partial manipulation, and a full manipulation is also a manipulation of any other type.

The general definition of Partial manipulation has been studied in [NP10, DL07]. The Hamming manipulation has been studied for a specific instance of classification in [MPR09]. A geometric definition similar to the Hamming manipulation has been studied in the context of facility location ([AFPT10]).

[4]In this version of the paper we will only refer to the uniform weight function. In any case, any weight function can be simulated via duplicate issues and uniform weights.

3 Partial Manipulation

3.1 Motivation

When there are no assumptions on the preferences of voters, we fear that any possible type of manipulation may be considered profitable by any of the voters. In that case, a strategy-proof aggregator must be partial-manipulation-free (PMF), as every manipulation is a partial manipulation. This definition of manipulation and the results in this section were introduced and discussed in previous works. Nehring and Puppe (2002, in ([NP10]), in a different context, arrived at a similar definition and the corresponding results. Dietrich and List [DL07] were the first to introduce this definition and theorems to the current context of judgment aggregation.

Since this is the broadest definition of manipulation, being immune to it is difficult, and the main results are impossibility theorems regarding the construction of PMF aggregators.

3.2 Impossibility Theorem

The property of being PMF gives rise to impossibility theorems under certain conditions on X, due to its connection to the property of being IIA, as stated in the following theorem:

Theorem 3.1: *[NP10, DL07] For all nonempty evaluation spaces $X \subseteq \{0,1\}^m$, an aggregator $f : X^n \to X$ is PMF if and only if it is IIA and monotone*[5] *.*

The notion of IIA aggregators is well studied, and there is a full characterization of the evaluation spaces X for which there is an impossibility theorem. The main property in this context is called *Totally Blocked*, which we will not define here[6].

The impossibility theorem for PMF aggregators is:

Theorem 3.2: *([NP10]) Every monotone and IIA aggregator $f : X^n \to X$ is dictatorial, if and only if an evaluation space $X \subseteq \{0,1\}^m$ is Totally Blocked.*

This theorem, combined with theorem 3.1 yields the following characterization of the cases for which there exists a a non-dictaroial PMF-aggregator:

Corollary 3.3: *([NP10, DL07]) Every PMF aggregator $f : X^n \to X$ is dictatorial, if and only if an evaluation space $X \subseteq \{0,1\}^m$ is Totally Blocked.*

4 Full manipulation

As was shown, designing an aggregator that is immune to partial manipulations is not always possible. In that case, we may still like to prevent weaker types of manipulation, with the weakest being full manipulation. More over, a manipulation-free aggregator under any type of definition is also full-manipulation free. Therefore, understanding the space of full-manipulation-free (FMF) aggregators is helpful in the design of manipulation-free aggregators under other definitions.

In this section we describe a set of aggregators which are FMF and also minimalize the cases in which there is a partial manipulation.

The natural question that comes up is what are the conditions on X such that there exists a FMF aggregator that is not dictatorial. It turns outs that for any set set X there

[5]The same theorem and proof hold for the general case where the range of f is a larger subset of $\{0,1\}^m$, i.e. $f : X^n \to Y$ and $X \subseteq Y \subseteq \{0,1\}^m$

[6]Due to size restrictions, we cannot give a full survey of the literature. The definition of *Totally Blocked* and the proof of theorem 3.2 can be found also in [DH10], which uses similar notation to this paper.

are such functions. We shall show an easy construction of aggregators that are FMF and not strictly dictatorial, but are still very far from being anonymous.

However, such aggregators are not as interesting, as the number of influential voters in such a scheme is independent on n. We shall focus more on the construction of an anonymous FMF aggregator, and show that it is also possible for every evaluation space X.

4.1 Partitions of Issues

We design a family of non dictatorial FMF aggregators based on a partition of the set of issues J to the set of voters, called *partition aggregators*. Consider the folowing partition of m issues into n subsets, $K = K_1 \cup K_2 \cup ... \cup K_n$ where $K_i \cap K_j = \emptyset$, (it is possible that some of the voters won't have any influence, i.e $K_i = \emptyset$). W.l.o.g. we will assume that $K_1 = \{1, 2...t_1\}, K_2 = \{t_1 + 1, ..., t_2\}, ...K_n = \{t_{n-1} + 1...t_n\}$. We go over the issues sequentially. The decision on issue $i \in K_j$ will follow the opinion of voter j unless the resulting partial evaluation on the issues $1, ..., i-1, i$ is infeasible. Formally, we define the social aggregator $f : X^n \to X$ inductively over i going from 1 to m to be:

$$(f(x)_i)_{i \in K_1} = x_i^1$$

$$\text{and} \quad (f(x)_i)_{i \in K_j} = \begin{cases} x_i^j & \text{the partial evaluation } (f(x)_1, ...f(x)_{i-1}, x_i^j) \text{ is feasible} \\ 1 - x_i^j & \text{otherwise} \end{cases}$$

The aggregator is consistent as a result of the inductive construction. The aggregator is a FMF aggregator since an agent j can change the result on issue i only by changing the result in at least one other issue in which his opinion was accepted. Therefore we get the following proposition

Proposition 4.1: *For every $X \subseteq \{0,1\}^k$, any partition aggregator is a FMF aggregator.*

A particularly interesting example is the *almost dictatorial* aggregator, obtained by taking $K_1 = \{1, ..., m-1\}, K_2 = \{m\}$

$$f(x) = \begin{cases} (x_1^1, ..., x_{m-1}^1, x_m^2) & (x_1^1, ..., x_{m-1}^1, x_m^2) \in X \\ x^1 & \textit{otherwise} \end{cases}$$

Note that the almost dictatorial aggregator is non manipulable, not only for this weak definition, but also for the weighted Hamming definition, for every X, when the issue determined by the second voter is the issue with the minimal weight. This means that there can be no impossibility theorem in the flavour of GS for the weighted Hamming manipulation.

Of course, it is not necessarily PMF for every X. there can be cases where it is beneficial for the voter deciding on the first $m-1$ issues to lie in order to gain on the m'th issue, by denying the second voter his influence.

4.2 Anonymous FMF Aggregators

An important approach for designing FMF aggregators is based on PMF aggregators. A basic property for a society would have to be to avoid partial manipulations whenever it is possible. From 3.1 we get that an aggregator is PMF if and only if is IIA and monotone. This fact is true not only for a consistent aggregator $f : X^n \to X$ but also for an aggregator from $X^n \to \{0,1\}^m$. As we saw in the impossibility theorem an IIA and monotone aggregator does not always produce outputs consistent with the evaluation space X. Therfore, we would like to correct these functions in the places where they are not consistent. We would like to study the set of aggregators which are consistent and yet "close" to an IIA and monotone aggregator.

Formally, for an inconsistent function $g : X^n \to \{0,1\}^m$, a consistent function $f : X^n \to X$ is called a correction of g if $f(x) = g(x)$ whenever $g(x)$ is consistent. Denote by $\mathbb{M}$ the set of all IIA and monotone functions $f : X^n \to \{0,1\}^m$. When f is a correction of a function $m \in \mathbb{M}$, at least all pairs of inputs for which m falls into X do not form a partial manipulation. We shall denote by $\mathbb{F}$ the set of consistent functions which are a correction of a function in $\mathbb{M}$. The functions in $\mathbb{F}$ will be called *close to partial manipulation free* aggregators (C-PMF).

Our aim in this chapter is to build a FMF-aggregator f with the property of being a C-PMF aggregator. We shall define the subset $\mathbb{G}$ of $\mathbb{F}$ to be the set of functions who are a composition of a function $g : \{0,1\}^m \to X$ with a function $m \in \mathbb{M}$ such that for every feasible evaluation $x \in X$, $g(x) = x$. Being a member of $\mathbb{G}$ means that the 'correction' part of the social aggregator in the cases where m, the IIA and Monotone stage, is not consistent, depends only on the outcome of m and not on the entire on the whole profile.

A special subset $\mathbb{H}$ of $\mathbb{G}$ is a composition of a Hamming nearest neighbour function h with a function $m \in \mathbb{M}$. A Hamming nearest neighbour function $h : \{0,1\}^m \to X$ is a function that, given $x \in \{0,1\}^m$, returns a closest point in X, under a given Hamming metric, i.e each issue has a nonzero weight[7]. Of course, such a function is not properly defined without a tie-breaking rule. We need to set proper tie-breaking rules in order to avoid manipulations. The main property we wish to maintain is that, given a nearest neighbour function h, if two different points $a, b \notin X$ both have the points $\alpha, \beta \in X$ in their set of potential nearest neighbours according to h, then it can not be that $h(a) = \alpha$ and $h(b) = \beta$.

One way of implementing that property is by choosing according to some lexicographical order in case of a tie. We shall denote the set of functions using the lexicographical tie-breaker as $\mathbb{H}^{\mathbb{1}}$ and the set of functions satisfying the aforementioned property as $\mathbb{H}^{\mathbb{2}}$, so:

$$\mathbb{H}^{\mathbb{1}} \subseteq \mathbb{H}^{\mathbb{2}} \subseteq \mathbb{H} \subseteq \mathbb{G} \subseteq \mathbb{F}$$

. We shall use the following notation in order to present the geometric relations of binary vectors a, b, c. We say that c is *between* a, b if for every coordinate i $a_i \leq c_i \leq b_i$ or $b_i \leq c_i \leq a_i$. The notation $[a, b]$ will describe the set of all the vectors between a and b $[a, b] = \{v | if\ a_i = b_i\ than\ v_i = a_i\}$. Likewise, (a, b) describes the set $[a, b] \backslash \{a, b\}$ and $[a, b) = [a, b] \backslash \{b\}$, etc. We say that $a \in X$ is a *neighbour* of $b \notin X$ if $(a, b) \cap X = \emptyset$

Theorem 4.2: *For every $X \subseteq \{0,1\}^k$, any social aggregator $f = h \circ m \in \mathbb{H}^{\mathbb{2}}$ ($m \in \mathbb{M}$), f is a FMF aggregator. Furthermore, if m is anonymous, then f is anonymous.*

Theorem 4.2 does not hold for any function in $\mathbb{F}$. Even if we use a function in $\mathbb{G}$ and the correction is done by choosing a neighbour which is not necessarily a nearest neighbour, then the aggregator is not necessarily FMF.

4.2.1 Social welfare maximizer

An important concept in mechanism design in the *social welfare maximizer*. Each individual in the society has a function returning his welfare given his opinion and the aggregated opinion. A social welfare maximizer is an aggregator that always returns the evaluation that maximizes the total welfare of all individuals in the society.

We consider the case where the welfare of every individual i is $-d_\omega(x^i, f(\mathbf{x}))$, the Hamming distance between his opinion and society's opinion, according to some weight function

[7]By definition any metric must maintain the following properties: non-negativity, idendity of indiscernibles, symmetry and the triangle inequalility. It is easy to check that a weighted Hamming distance is a metric if and only if each issue has a nonzero weight.

with positive weights on all the issues. The corresponding social welfare maximizer is the function

$$f(\mathbf{x}) = argmin_{x\in X} \sum_{i\in N} d_\omega(x, x^i)$$

We call this aggregator f a Hamming social welfare maximizer. In case of a tie, we shall use a tie-breaking rule which will ensure that $f \in \mathbb{H}^2$. Notice that $f \in \mathbb{F}$ since it is the correction of the IIA and monotone aggregator $\tilde{f} : X^n \to \{0,1\}^k$ where

$$\tilde{f}(\mathbf{x}) = argmin_{x\in\{0,1\}^k} \sum_{i\in N} d_\omega(x, x^i)$$

However, $f \notin \mathbb{G}$, because the correction depends on the entire profile. It is easy to construct two profiles $\mathbf{x}, \mathbf{y}$ such that $\tilde{f}(\mathbf{x}) = \tilde{f}(\mathbf{y})$ and $f(\mathbf{x}) \neq f(\mathbf{y})$ [8].

We shall show that this aggregtor has the same property of being FMF.

Theorem 4.3: *For every evaluation space $X \subseteq \{0,1\}^k$, a Hamming social welfare maximizer is FMF and anonymous.*

The Hamming social welfare maximizer has been used before. In preference aggregation, it is known as Kemeny's rule ([LY78]). There are many works that discuss various aspects of Kemeny's rule[9], but not in connection with strategy-proofness, as far as we know. facility location [AFPT10], classfication [MPR09] and more [Pig06]. A general connection between social welfare maximization and strategy-proofness was not previously known.

5 Hamming Manipulations

5.1 Main Results

In this section we present some results regarding the Hamming manipulation. We say that voter i with opinion x prefers the result $v \in X$ more than $u \in X$ if the distance, according to a weighted Hamming metric ω, $d_\omega(x, v)$ of v from x is less than the distance of u from x.

As was mentioned in the previous chapter, the almost dictator aggregator is HMF for any weighted hamming definition. Therefore, we will focus on the interesting case of building an anonymous HMF aggregator. Following the results of the previous chapter, we focus on the set of aggregators $\mathbb{H}$. We show two conditions for determining whether an aggregator $f \in \mathbb{H}$ is not only FMF, but also HMF. We shall discuss the cases in which such an aggregator is non-HMF. More over, we use these two lemmas to analyze some special cases and show whether there is an HMF aggregator in $\mathbb{H}$.

We show that in any case where there is a manipulation of an aggregator $h \circ m \in \mathbb{H}$ on the profile $\mathbf{x}$, the 2 intermediate results $w = m(x^i, x^{-i})$ and $z = m(y, x^{-i})$ must both be outside of X, not too "far" from each other (lemma 5.4) and not too "close" to each other (lemma 5.5). For that we will use a combinatorial representation of the evaluation space X.

Definition 5.1: For a non-empty evaluation space $X \subseteq \{0,1\}^m$, A minimally infeasible partial evaluation (abbreviated MIPE) is a K-evaluation $x = (x_j)_{j\in K}$ for some $K \subseteq J$ which is infeasible, but such that every restriction of $\mathbf{x}$ to a proper subset of K is feasible.

[8] For example let $X = \{110000, 001000, 000111\} \subseteq \{0,1\}^6$ and $n = 9$. $\mathbf{x}$ will be the profile where 3 agents hold the first opinion 110000, 2 agents hold the second opinion 001000 and 4 agents hold the last opinion 000111. $\mathbf{y}$ will be the profile where 3 agents hold the first opinion,3 agents hold the second one and 3 agents hold the last one. By taking the uniform weights we get that $f(\mathbf{x}) = 000111$, $f(\mathbf{y}) = 001000$ and $\tilde{f}(\mathbf{x}) = \tilde{f}(\mathbf{y}) = 000000$

[9] see, for example [ACN08]

X can be defined by its set of MIPEs. A MIPE represents a maximal Boolean subcube that is outside of X[10].

Definition 5.2: For every MIPE $a = (a_j)_{j\in K}$ we denote by T_a, the *MIPE-set* of a, as the following subset of $\{0,1\}^m$: $T_a = \{x|x_{|_K} = a\}$.

Definition 5.3: For every evaluation $x \in X^c$, we denote by $MT(x)$, its *MIPE-type* as the following set of MIPES of X: $MT(x) = \{a|x \in T_a\}$

We are now prepared to bring the partial characterizations for the general case:

Lemma 5.4: *For every $X \subseteq \{0,1\}^k$, and any social aggregator $f \in \mathbb{H}$, $f = h_\omega \circ m$, if y is a ω-Hamming manipulation of i over (x^i, x^{-i}), then $[(m(x^i, x^{-i})), (m(y, x^{-i}))] \cap X = \emptyset$. (In other words there exists an MIPE a such that $(m(x^i, x^{-i})), (m(y, x^{-i})) \in T_a$.)*

Lemma 5.5: *For every $X \subseteq \{0,1\}^k$, and any social aggregator $f \in \mathbb{H}$ $f = h_\omega \circ m$, if y is a ω-Hamming manipulation of i over (x^i, x^{-i}), then $MT(m(x^i, x^{-i})) \neq MT(m(y, x^{-i}))$.*

5.2 examples

These two theorems do not give a full characterization for the sets X for which there exists a manipulation free aggregator. However, they show that for aggregators in $\mathbb{H}$, a Hamming manipulation occurs only in special circumstances. For many particular cases, including the preference aggregation model, we can conclude whether or not there exists an HMF aggregator in $\mathbb{H}$. In this subsection We shall present for two particular cases[11] the preference model and the "k choose m" model, to be defined later on.

We shall show that for the preference aggregation model, when there are three alternatives any combination of a monotone aggregator and the standard nearest neighbor aggregator is an HMF aggregator but not for more than three alternatives. A general natural question that arises (and is still open) is what is the minimal number of alternatives for which there is no anonymous HMF and C-PMF aggregator.

For the "k choose m" decision example we shall present some anonymous HMF C-PMF aggregators for any number k and m. Those examples will give us some intuition regarding the existence of HMF aggregators and the usage of the Theorems.

5.2.1 Preference Aggregation

We shall denote the set of alternatives by $A = \{a, b, c, ...\}$, $|A| = k$. The set of issues K is the set of pairwise preferences between every 2 alternatives , so $m = \binom{k}{2}$. For $k > 2$ it is well known from Arrow's theorem that there is no IIA and Monotone aggregator and therefore there isn't a PMF aggregator. In the next claim we shall show that there is an anonymous, HMF and C-PMF aggregator for three alternatives.

Claim 5.6: If $m = 3$, then all aggregators $h_\omega \circ m$ in $\mathbb{H}$ are ω-HMF.

For more than three alternatives we will not bring a full answer to the question of whether there exists an anonymous HMF and C-PMF aggregators and we will show that it can't be of the form $h_\omega \circ m$. [12]

Claim 5.7: In preference aggregation over at least $k \geq 4$ alternatives, and at least 3 voters, aggregators $h \circ maj \in \mathbb{H}$ are not HMF.

[10] An IIA and monotone aggregator over X is anonymous, neutral, PMF and consistent iff all its MIPES are of size 2 [NP10]

[11] The cases of facility location on a line and a cycle are shown in a subsequent work.

[12] In another work in which we use random functions it can be shown that there exists an HMF aggregator $h_\omega \circ m$ for four alternatives, where h is random and m is monotone.

References

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Mathematics of Operations Research*, 55(5), 2008.

[AFPT10] Noga Alon, Michal Feldman, Ariel D. Procaccia, and Moshe Tennenholtz. Strategyproof approximation of the minimax on networks. *Mathematics of Operations Research*, 35(3):513–526, 2010.

[Arr63] Kenneth J. Arrow. *Social Choice and Individual Values.* John Wiley and sons, New York, 1963.

[DFMN12] Elad Dokow, Michal Feldman, Reshef Meir, and Ilan Nehama. Strategyproof classification with shared inputs. In *13th ACM Conference on Electronic Commerce(EX '12)*, Valencia, Spain, 2012.

[DH10] Elad Dokow and Ron Holzman. Aggregation of binary evaluations. *Journal of Economic Theory*, 145:495–511, 2010.

[DL07] Franz Dietrich and Christian List. Strategy-proof judgment aggregation. *Economics and Philosophy*, 23:269–300, 2007.

[Gib73] Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41(4):587–601, 1973.

[LP10] Christian List and Ben Polak. Introduction to judgment aggregation. In *Cowles Foundation Discussion Paper No. 1753*, 2010.

[LY78] Arthur Levenglick and Hobart P. Young. A consistent extension of condorcet's election principle. *SIAM Journal on Applied Mathematics*, 35(2):285–300, 1978.

[MPR09] Reshef Meir, Ariel D. Procaccia, and Jeffrey S. Rosenschein. Strategyproof classification with shared inputs. In *The Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pages 220–225, Pasadena, California, July 2009.

[NP10] Klaus Nehring and Clemens Puppe. Abstract arrowian aggregation. *Journal of Economic Theory*, 145:467–494, 2010.

[Pig06] G. Pigozzi. Belief merging and the discursive dilemma: an argument-based account to paradoxes of judgment aggregation. *Synthese*, 152(2):285–298, 2006.

[Sat75] Mark A. Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, April 1975.

Dvir Falik
Tel-Aviv University, Israel and Microsoft Research, Herzlia, Israel
Email: dvir.falik@mail.huji.ac.il

Elad Dokow
Bar Ilan University, Israel and Carlos III de Madrid university, Spain
Email: eladokow@gmail.com

Bounded single-peaked width and proportional representation[1]

Denis Cornaz, Lucie Galand and Olivier Spanjaard

Abstract

This paper is devoted to the proportional representation (PR) problem when the preferences are clustered single-peaked. PR is a "multi-winner" election problem, that we study in Chamberlin and Courant's scheme [6]. We define clustered single-peakedness as a form of single-peakedness with respect to clusters of candidates, i.e. subsets of candidates that are consecutive (in arbitrary order) in the preferences of all voters. We show that the PR problem becomes polynomial when the size of the largest cluster of candidates (width) is bounded. Furthermore, we establish the polynomiality of determining the single-peaked width of a preference profile (minimum width for a partition of candidates into clusters compatible with clustered single-peakedness) when the preferences are narcissistic (i.e., every candidate is the most preferred one for some voter).

1 Introduction

Social choice theory deals with making collective choices on the basis of the individual preference relations of a set of agents (or voters) over a set of alternatives (or candidates). In this field, an active stream of research deals with "multi-winner" elections, where one aims at electing a subset of candidates rather than a single candidate. This occurs for instance when electing an assembly. In such situation, a combinatorial difficulty arises: while there are only m possible outputs of a single-winner election with m candidates, there are $\binom{m}{\kappa}$ possible assemblies of κ representatives. This difficulty is often overcome by organizing κ single-winner elections over κ subelectorates. With this way of partitioning the election, it may nevertheless happen that the elected assembly fails to represent minorities [4]: assume that the representatives of a party are in second position for the κ single-winner elections, then the party will have no representative in the assembly. Proportional representation aims at tackling this issue by performing a single multi-winner election ensuring that collectively the voters are satisfied enough by at least one elected candidate. This can be achieved for instance by using Chamberlin and Courant's scheme [6], where one elects a subset of κ candidates minimizing a *misrepresentation* score. The effective computation of such winning subsets of candidates has been studied by several authors.

Procaccia et al. have shown that the problem is NP-hard in the general case, but polynomial for a fixed κ [12]. Lu and Boutilier provided a polynomial approximation algorithm with performance guarantee (for maximizing a *representation* score), and show, on different experimental datasets, that it almost always returns an optimal solution [10]. Their setting is nevertheless different from proportional representation in political science: they aim at designing a system able to recommend a set of options to a group, based on the individual preferences of its members. Such a system could be used for instance by a conference organizer wishing to select a subset of sushis for the gala dinner, based on the individual preferences of the participants over the varieties of sushis. Clearly, this context authorizes suboptimality. Coming back to voting procedures, it is nevertheless important to note that the scores only provide an ordinal information: if an assembly A has a misrepresentation score 1 while an assembly B has a misrepresentation score $1 + \varepsilon$, one can only conclude that A is better than B, and not that B is close to be as good as A. Furthermore, in a political setting, it is simply not possible to elect an assembly without guaranteeing that it is the true winner. To our

[1]This paper appeared in the proceedings of ECAI 2012.

knowledge, the only general exact approaches proposed for proportional representation in Chamberlin and Courant's scheme are based on integer programming as the ones by Potthoff and Brams [11] and by Balinski [1] (in this latter reference, the formulation was actually proposed for the κ-median location problem, which is equivalent to the proportional representation problem [12]). The solution of these IP formulations might of course take exponential time in the worst case.

Very recently, Betzler et al. proposed an extensive investigation of parameterized complexity results for the problem [4]. Besides, they established that the problem becomes polynomial when the preferences are *single-peaked* [5]. Single-peakedness is the most popular domain restriction in social choice theory. In single-winner elections, it makes it possible to overcome Arrow's impossibility theorem (that states that no voting rule can simultaneously fulfill a set of basic axioms). In particular, there always exists a Condorcet winner (i.e., a candidate who is preferred to any other candidate by a majority of voters) if preferences are single-peaked. Such preferences are typically encountered in political science. Intuitively, preferences are single-peaked when 1) all voters agree on a left-right axis on the candidates reflecting their political convictions, and 2) the preferences of all voters decrease along the axis when moving away from their preferred candidate to the right or left. Nevertheless, this condition on preferences can be a bit restrictive when several candidates share similar opinions (e.g. they belong to the same party) since it is unlikely that the preferences of all voters are single-peaked on this subset of candidates.

We therefore study a new domain restriction, *clustered single-peakedness*, where single-peakedness holds on subsets of candidates (parties or more generally clusters), and not within clusters. The candidates belonging to the same cluster are ranked consecutively in the preferences of all voters, though *not necessarily in the same order*. Given a partition of the candidates into clusters such that the preferences are clustered single-peaked, the *width* of the partition is the size of the largest cluster minus one. Note that, for a given set of individual preference relations, several partitions into clusters can be compatible with clustered single-peakedness: we call *single-peaked width* the minimum width among all possible partitions of candidates into clusters. We show that the single-peaked width is computable in polynomial time if preferences are narcissistic, and that a bounded single-peaked width makes it possible to design a polynomial time solution algorithm for the proportional representation problem. Note that the same structures have been studied by Elkind et al. [7], under another terminology (in particular, clusters are called clone sets). Their main concern is not to study how clustered single-peakedness can be used to determine the winner of an election, but they show interesting connections with PQ-trees, and use them to design an algorithm to compute a partition of the candidates into (as many as possible) clusters. The links between their work and ours will be detailed in Section 4.

The paper is organized as follows. We first formally introduce the proportional representation problem and clustered single-peakedness (Sect. 2). Then we present a dynamic programming procedure for solving the proportional representation problem when the preferences are clustered single-peaked (Sect. 3). A key parameter for the efficiency of the procedure is the width of the partition into clusters. We therefore study the complexity of determining the single-peaked width of a set of individual preference relations (Sect. 4), and show the polynomiality of the problem for *narcissistic* preferences.

2 Preliminaries

2.1 Proportional representation

Let V be a set of n voters and C a set of m candidates. Let $\mathcal{P}$ be an $m \times n$ *preference profile matrix* over C, that is, each candidate appears exactly once in each column. So the set of columns of $\mathcal{P}$ is the set V and each column v is the preference relation of voter v. We denote by $r(v, c)$ the rank of candidate c in the preferences of voter v, and by $x \prec_v y$ the preference for y over x. A non-decreasing misrepresentation function $\mu : \{1, \ldots, m\} \to \mathbb{N}$ is defined such that

$\mu(r(v, c))$ is the misrepresentation value of c for v. The proportional representation problem aims at determining a subset $S \subseteq C$ of κ candidates such that the total misrepresentation score is minimized. In Chamberlin and Courant's scheme, the scoring function $s : 2^C \rightarrow \mathbb{N}$ is defined as follows:

$$s(S) = \sum_{v \in V} \min_{c \in S} \mu(r(v, c))$$

The proportional representation problem can then be simply written: $\min_{|S|=\kappa} s(S)$. The following example illustrates the value of using Chamberlin and Courant's scheme.

Example 1. *Consider a proportional representation problem with 6 voters* $1, 2, 3, 4, 5, 6$ *(indices of the columns) and 4 candidates* a, b, c, d*, and the following preference profile matrix:*

$$\mathcal{P} = \begin{pmatrix} a & c & a & c & d & c \\ b & b & b & a & c & d \\ c & a & c & b & b & a \\ d & d & d & d & a & b \end{pmatrix}$$

Assume that the misrepresentation function is $\mu(r) = r - 1$*. If* $\kappa = 2$*, then the possible subsets and scores are (for simplicity* ab *stands for* $\{a, b\}$*):*

ab	ac	ad	bc	bd	cd
6	1	4	3	6	4

The optimal solution is subset ac *with score 1. With such a solution, only one voter is not represented by her preferred candidate (but by her second choice). Assume now this multi-winner election is divided into two single-winner elections, namely an election* L *between* b *and* c *for voters* $1, 2, 3$*, and an election* R *between* a *and* d *for voters* $4, 5, 6$*. The winner of election* L *(resp.* R*) is* b *(resp.* d*). Consequently, the winning solution is* bd*, which is the worst one according to the misrepresentation scores!*

2.2 Clustered single-peakedness

Definition 1. *Let* $\mathcal{C} = (C_1, \ldots, C_q)$ *be an ordered partition of* C *into* q *non-empty subsets (called* clusters*). Preference profile matrix* $\mathcal{P}$ *is* clustered single-peaked *with respect to* $\mathcal{C}$ *if for all* $v \in V$ *there exists an index* p *in* $\{1, \ldots, q\}$ *such that:*

$$i < j < p \quad \Rightarrow \quad x \prec_v y \prec_v z$$
$$p < j < i \quad \Rightarrow \quad x \prec_v y \prec_v z$$

for all $x \in C_i$*,* $y \in C_j$ *and* $z \in C_p$*.*

For a voter v, we call C_p the *peak* of v, which means that any candidate in C_p is preferred to any candidate in $C \setminus C_p$. This definition coincides with usual single-peakedness when $|C_i| = 1$ for all i. The only candidate in C_p is then the most preferred one.

Example 2. *Coming back to Example 1, it can be easily seen that the preferences are not single-peaked w.r.t. axis* (a, b, c, d)*, by considering Figure 1 where each curve represents a preference ranking of a voter, namely voters 1, 2, 6. For each curve and each candidate on the X-axis, the value on the Y-axis is the rank in the corresponding preference ranking (the better the rank the higher the point). Preferences are single-peaked w.r.t. an X-axis iff all curves have a single peak. This is not the case in the left graph since the curve of voter 6 (in bold) spikes down for* b *and then spikes up for* a*. More generally, it can be shown that the preferences in Example 1 are not single-peaked, whatever permutation of candidates on the X-axis is considered. However, the preferences are clustered single-peaked with respect to* $(\{a, b\}, \{c\}, \{d\})$*, denoted by* (ab, c, d) *for simplicity. Note that* a *and* b *are adjacent in all preference rankings, which is a necessary condition to be clustered (but not sufficient for clustered single-peakedness!). A preference profile is clustered single-peaked with respect to an ordered partition* $(C_1, \ldots, C_q)$ *iff it is single-peaked when considering each subset* C_i *as a single candidate. In the example, introducing cluster* $\{a, b\}$ *amounts to considering* a *and* b *as a "single candidate"* ab*. The preference profile matrix becomes then the one indicated on the right-hand side of Figure 1. In the graph on the right, one can observe that the preferences become then single-peaked, i.e. they are clustered single-peaked with respect to* (ab, c, d)*.*

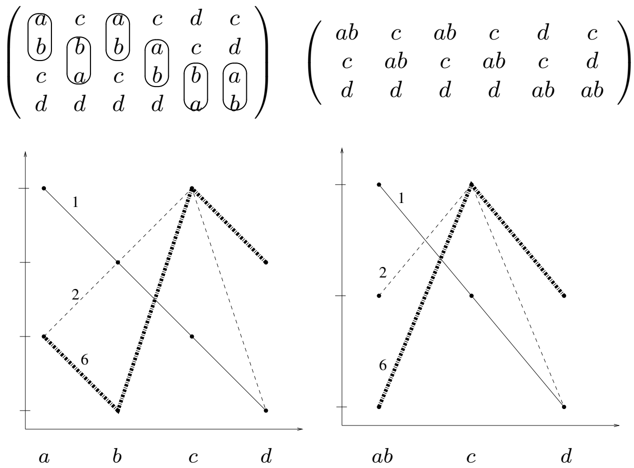


Figure 1: Clustered single-peakedness.

3 Dynamic Programming

We now present a dynamic programming algorithm that generalizes the one proposed by Betzler et al. for single-peaked preferences [4]. Let $P(i, C', k)$ denote the subproblem where all candidates in $C' \subseteq C$ are made mandatory and one selects $k - |C'|$ candidates in $C_1 \cup \ldots \cup C_i$. For the convenience of the reader, we briefly recall the recursion scheme of the procedure proposed by Betzler et al., with an alternative proof. Assume that the preferences are single-peaked with respect to axis $(x_1, \ldots, x_m)$ (i.e. clustered single-peaked with respect to $(C_1, \ldots, C_m)$, where $C_i = \{x_i\}$ $\forall i$). Let $z(i, k)$ denote the optimal score for problem $P(i-1, \{x_i\}, k)$, where one selects x_i and $k-1$ candidates among $\{x_1, \ldots, x_{i-1}\}$ (the $i-1$ leftmost candidates on the axis). The authors use the following recursion:

$$z(i,k) = \min_{j \in [k-1 \cdot\cdot i-1]} \left\{ z(j, k-1) - \sum_{v \in V} \max\{0, \mu(r(v, x_j)) - \mu(r(v, x_i))\} \right\}$$

The optimal score for a subset of κ candidates is then $\min_{i \in \{\kappa \cdot\cdot m\}} z(i, \kappa)$. The validity of the recursion can be established by showing that selecting a subset of k candidates, including x_j and x_i (mandatory candidates), in $\{x_1, \ldots, x_j, x_i\}$ (problem $P(j-1, \{x_j, x_i\}, k)$) amounts to selecting $k-1$ candidates, including x_j, in $\{x_1, \ldots, x_j\}$ (problem $P(j-1, \{x_j\}, k-1)$). Indeed, it reduces to computations on the same *minor* of the preference profile.

Definition 2. *Any preference profile matrix that depicts the individual preferences of a subset $V' \subseteq V$ of voters over a subset $C' \subseteq C$ of candidates is called a* minor *and denoted by $\mathcal{P}(V', C')$.*

The voters can be partitioned into two sets: the set $V_{[1,j-1]}$ of voters whose peak x_p is in $\{x_1, \ldots, x_{j-1}\}$, and the set $V_{[j,m]}$ of voters whose peak x_p is in $\{x_j, \ldots, x_m\}$. Both problems $P(j-1, \{x_j, x_i\}, k)$ and $P(j-1, \{x_j\}, k-1)$ amount to computations in the same minor:

• *Problem* $P(j-1, \{x_j, x_i\}, k)$: all voters in $V_{[j,m]}$ can be deleted from the preference profile matrix since their preferred candidate among $\{x_1, \ldots, x_j, x_i\}$ is either x_i or x_j, that are mandatory, and therefore the preferences of these voters play no role in the determination of the optimal solution to $P(j-1, \{x_j, x_i\}, k)$. Furthermore, all voters in $V_{[1,j-1]}$ prefer x_j to x_i since their peak is to the left of x_j, and therefore candidate x_i plays no role since x_j is mandatory. Consequently, the problem reduces to selecting $k-1$ candidates, including x_j, according to minor $\mathcal{P}(V_{[1,j-1]}, \{x_1, \ldots, x_j\})$.

• *Problem* $P(j-1, \{x_j\}, k-1)$: for all voters in $V_{[j,m]}$, candidate x_j is necessarily the most preferred one in $\{x_1, \ldots, x_j\}$. Since candidate x_j is mandatory, all voters in $V_{[j,m]}$ can be deleted from the preference profile matrix. The problem reduces then to selecting $k-1$ candidates, including x_j, according to minor $\mathcal{P}(V_{[1,j-1]}, \{x_1, \ldots, x_j\})$.

The two problems $P(j-1, \{x_j, x_i\}, k)$ and $P(j-1, \{x_j\}, k-1)$ are thus equivalent, which establishes the validity of the recursion. We now show how this recursion scheme can be extended to handle clustered single-peaked preferences. Assume that the preferences are clustered single-peaked with respect to an ordered partition $(C_1, \ldots, C_q)$. Let $z(i, C'_i, k)$ denote the optimal score when candidates in $C'_i \subseteq C_i$ are mandatory, candidates in $C_i \setminus C'_i$ are forbidden, and one selects $k - |C'_i|$ candidates in $C_1 \cup \ldots \cup C_{i-1}$. In our setting, the recursion can be written as follows:

$$z(i, C'_i, k) = \min_{j \in [1 \cdots i-1]} \min_{C'_j \subseteq C_j \, C'_j \neq \emptyset} \left\{ z(j, C'_j, k - |C'_i|) - \sum_{v \in V} \max\{0, \min_{y \in C'_j} \mu(r(v,y)) - \min_{x \in C'_i} \mu(r(v,x))\} \right\} \quad (1)$$

where $z(i, C'_i, k) = +\infty$ if $|C'_i| > k$ or $|C_1 \cup \ldots \cup C_{i-1}| < k - |C'_i|$.
The optimal score for a subset of κ candidates is then:

$$\min_{i \in [1 \cdots q]} \min_{C'_i \subseteq C_i \, C'_i \neq \emptyset} z(i, C'_i, \kappa)$$

The proof of the recursion is similar to the one in the single-peaked case. It amounts to establishing the equivalence of problems $P(j-1, C'_j \cup C'_i, k)$ and $P(j-1, C'_j, k - |C'_i|)$, by considering a partition of V into the set $V_{[1,j-1]}$ of voters whose peak is in $\{C_1, \ldots, C_{j-1}\}$ and the set $V_{[j,q]}$ whose peak is in $\{C_j, \ldots, C_q\}$:

• *Problem* $P(j-1, C'_j \cup C'_i, k)$: all voters in $V_{[j,q]}$ can be deleted from the preference profile matrix since their preferred candidate among $C_1 \cup \ldots \cup C_{j-1} \cup C'_j \cup C'_i$ is either in C'_i or in C'_j. All voters $V_{[1,j-1]}$ prefer a candidate in C'_j to a candidate in C'_i since their peak is to the left of C_j. Consequently, the problem reduces to selecting $k - |C'_i|$ candidates, including candidates in C'_j, according to minor $\mathcal{P}(V_{[1,j-1]}, C_1 \cup \ldots \cup C_{j-1} \cup C'_j)$.

• *Problem* $P(j-1, C'_j, k - |C'_i|)$: for all voters in $V_{[j,q]}$, the most preferred candidate in $C_1 \cup \ldots \cup C_{j-1} \cup C'_j$ necessarily belongs to C'_j. The voters can therefore be deleted from the preference profile. The problem reduces then to selecting $k - |C'_i|$ candidates, including candidates in C'_j, according to minor $\mathcal{P}(V_{[1,j-1]}, C_1 \cup \ldots \cup C_{j-1} \cup C'_j)$.

Both problems are thus equivalent, which establishes the validity of the recursion. Algorithm 1 describes the ensuing dynamic programming procedure.

Algorithm 1: Dynamic programming

for $i = 1, \ldots, q$ **do**
  for $C'_i \subseteq C_i$ *with* $|C'_i| \leq \kappa$, $C'_i \neq \emptyset$ **do**
    $z(i, C'_i, |C'_i|) = \sum_{v \in V} \min_{x \in C'_i} \mu(r(v,x))$
for $i = 2, \ldots, q$ **do**
  for $C'_i \subseteq C_i$ *with* $|C'_i| \leq \kappa$, $C'_i \neq \emptyset$ **do**
    for $k = |C'_i| + 1, \ldots, \min\{\kappa, |C'_i| + \sum_{j=1}^{i-1} |C_j|\}$ **do**
      compute $z(i, C'_i, k)$ by Equation 1
return $\min_{i \in [1 \cdots q]} \min_{C'_i \subseteq C_i, \, C'_i \neq \emptyset} z(i, C'_i, \kappa)$

Example 3. *For simplicity, a set* $\{a, b\}$ *is denoted by ab in this example, and* $\{a, b\} \cup \{c, d\}$ *by abcd. Consider a proportional representation problem with 6 candidates* a, b, c, d, e, f *having clustered single-peaked preferences with respect to* (ab, cd, e, f). *Let us study how many triples of* *candidates are examined by the procedure when computing $z(4, f, 3)$. Given r subsets $S_1, \dots, S_r$ of candidates, let us denote by $\text{opt}\{S_1, \dots, S_r\}$ a subset in $\arg\min_i s(S_i)$. The following computation is performed by the procedure:*

$$z(4, f, 3) = s\left(\text{opt}\left\{\begin{array}{l} fab, f\text{opt}\{ca, cb\}, f\text{opt}\{da, db\}, \\ fcd, f\text{opt}\{ea, eb, ec, ed\} \end{array}\right\}\right)$$

Therefore 5 subsets are examined (three of the four "opt" operations have been performed during the previous iterations) while there are 10 subsets of cardinality 3 including f.

For a small single-peaked width, the computational savings become of course more and more significant when the size of the instance increases. Actually, the following complexity analysis shows that the dynamic programming procedure is polynomial for a bounded singe peaked width. Equation 1 requires indeed a computational time within $O(nqt2^t)$ where $t = \max_i |C_i| - 1$. Furthermore, the number of computed terms $z(i, C'_i, k)$ is upper bounded by $q2^t\kappa$. Therefore the running time of the procedure is within $O(nq^2t2^{2t}\kappa)$, which amounts to $O(nm^3)$ for a bounded single-peaked width t (we recall that $q \leq m$ and $\kappa \leq m$).

Theorem 1. *The proportional representation problem over bounded single-peaked width preferences is polynomial.*

The complexity analysis shows that $\max_i |C_i|$ is a key parameter for the efficiency of the algorithm. Note that there always exists an ordered partition for which the preferences are clustered single-peaked: in the worst case, it is sufficient to consider the partition (C). It is nevertheless interesting from an algorithmic viewpoint to have an ordered partition where each subset includes few candidates. Two cases can occur: either the partition is known in advance (for instance, when the candidates indicate their affiliation to a political party and the preferences of the voters are consistent with the displayed affiliations) or it is unknown. In both cases, it is desirable to be able to compute an ordered partition compatible with clustered single-peakedness and such that $\max_i |C_i|$ is minimized. In the next section, we show the polynomiality of this problem for *narcissistic preferences* [3, 13].

4 Single peaked width

We call *width* of an ordered partition $(C_1, \dots, C_q)$ the value $\max_i |C_i| - 1$. Given a preference profile matrix, we call *single-peaked width* the minimum width among all ordered partitions compatible with clustered single-peakedness. This can be seen as a distance measuring near-single-peakedness (the single-peaked width is indeed equal to 0 for single-peaked preferences). Note that this should not be confused with other distance measures that have been proposed in the literature, such as the number of voters to remove to make a profile single-peaked [9].

Example 4. *Consider the preference profile matrix $\mathcal{P}$ represented in Figure 2, where the preferences are not single-peaked. It is easy to check that they are nevertheless clustered single-peaked with respect to ordered partition (ac, efg, bd, h) (see the left part of the figure, where the subsets of the partition are encircled), whose width is $|\{e, f, g\}| - 1 = 2$. However the preferences are also clustered single-peaked with respect to (ac, f, eg, b, d, h) (right part of the figure). The single-peaked width of this preference matrix is thus 1.*

Ballester and Haeringer [2] recently showed that single-peakedness can be lost just because of the existence of two voters and four candidates, or three voters and three candidates. Conversely, they showed that if a profile is not single-peaked there must exist a set of two voters (resp. three) whose preferences over four candidates (resp. three) are not single-peaked. More precisely, the authors characterize single-peakedness with the following two conditions:

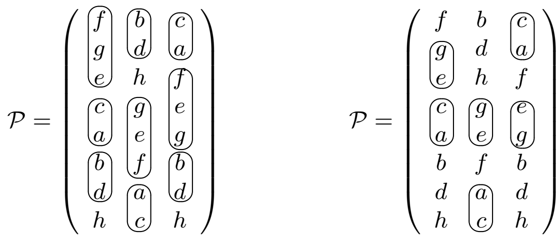


Figure 2: Single-peaked width.

• *Worst-restriction:* Given a triple $V' \subseteq V$ of voters and a triple $C' \subseteq C$ of candidates, let $L(V', C')$ be the set of all candidates ranked last in C' by at least one voter in V'. The worst-restriction condition holds if $|L(V', C')| < 3$ for all triples V' and C'.

• *α-restriction:* the α-restriction condition holds if there do not exist two voters v and v' and four candidates w, x, y, and z such that their preferences over w, x and z are opposite ($w \succ_v x \succ_v z$ and $z \succ_{v'} x \succ_{v'} w$) and the voters agree about the preference for y over x ($y \succ_v x$ and $y \succ_{v'} x$).

Interestingly, these conditions amount to forbidding five minors in the profile $\mathcal{P}$ (Lemma 1). In this formalism, we propose here a shorter proof of the characterization result of Ballester and Haeringer. Our proof is based on the polynomial algorithm proposed by Escoffier et al. [8] to determine if a profile is single-peaked with respect to some axis. This algorithm runs in time $O(mn)$ improving on the $O(mn^2)$ algorithm proposed by Bartholdi and Trick [3]. Before stating Lemma 1, let us present the algorithm of Escoffier et al. It works recursively and takes as arguments the left part $(x_1, \ldots, x_i)$ and the right part $(x_j, \ldots, x_m)$ of the axis under construction. A third argument is the subset C' of candidates which remains to be positioned on the axis. This algorithm returns an axis compatible with $\mathcal{P}$ or proves that the preferences are not single-peaked (by raising a contradiction between voters). The recursion is made possible by the fact that single-peakedness over $\mathcal{P}$ implies single-peakedness over any of its minors. It heavily uses the property that candidates ranked last in the preferences are necessarily at the extremities of the axis. At each step of the algorithm, one candidate x or two candidates x and y are ranked last in $\mathcal{P}(V, C')$ and will be positioned in x_{i+1} or x_{j-1} on the axis. There is a *contradiction* if a candidate has to be placed in two different positions (according to the preferences of two voters). These positions depend on the way x and y are positioned with respect to x_i and x_j in the preferences of all the voters. The whole procedure is detailed in Algorithm 2. The initial call is Make-axis(C, (), ()).

Before presenting Lemma 1 (on which our algorithm to compute single-peaked width strongly relies), we need to introduce the notion of *isomorphic* minors. A minor $\mathcal{P}'$ is isomorphic to $\mathcal{P}$ if there exists a bijection ϕ such that $\mathcal{P}$ and $\mathcal{P}'$ are identical up to column permutation if one renames every candidate x in $\mathcal{P}$ as $\phi(x)$. For instance, preference profile matrix $\mathcal{P}'$ below is isomorphic to $\mathcal{P}$ (take $\phi(a) = b$, $\phi(b) = c$, $\phi(c) = a$ and permute the columns).

$$\mathcal{P} = \begin{pmatrix} a & c \\ b & a \\ c & b \end{pmatrix}, \quad \mathcal{P}' = \begin{pmatrix} a & b \\ b & c \\ c & a \end{pmatrix}$$

Definition 3. *A minor is called* forbidden *if it is isomorphic to one of the following profiles:*

$$\mathcal{T}_1 = \begin{pmatrix} a & b & c \\ b & c & a \\ c & a & b \end{pmatrix}, \mathcal{T}_2 = \begin{pmatrix} a & c & a \\ b & b & c \\ c & a & b \end{pmatrix},$$
$$\mathcal{F}_1 = \begin{pmatrix} a & c \\ d & d \\ b & b \\ c & a \end{pmatrix}, \mathcal{F}_2 = \begin{pmatrix} a & d \\ d & c \\ b & b \\ c & a \end{pmatrix}, \text{ or } \mathcal{F}_3 = \begin{pmatrix} d & d \\ a & c \\ b & b \\ c & a \end{pmatrix}.$$

Algorithm 2: Make-axis(C',$(x_1,\ldots,x_i)$,$(x_j,\ldots,x_m)$)

1 **if** $C'=\emptyset$ **then return** $(x_1,\ldots,x_i,x_j,\ldots,x_m)$
2 **if** $C'=\{x\}$ **then return** $(x_1,\ldots,x_i,x,x_j,\ldots,x_m)$
3 $L \leftarrow$ candidates ranked last in $\mathcal{P}(C',V)$ by at least one voter
4 **if** $L=\{x\}$ **then** $y \leftarrow$ a candidate in $C'\setminus\{x\}$ /* $x \prec_v y, \forall v$ */
5 **if** $|L|\geq 3$ **then return** *not single-peaked*
6 **for** $v=1,\ldots,n$ **do**
7   **if** $L=\{x,y\}$ **then let** $x \prec_v y$ (w.l.o.g)
8   **if** $x_i \prec_v x \prec_v x_j \prec_v y$ **or** $x_i \prec_v x \prec_v y \prec_v x_j$ **then**
9   **if** *no contradiction* **then** $x_{i+1} \leftarrow x$; $x_{j-1} \leftarrow y$
10   **else return** *not single-peaked*
11   **if** $x_j \prec_v x \prec_v x_i \prec_v y$ **or** $x_j \prec_v x \prec_v y \prec_v x_i$ **then**
12   **if** *no contradiction* **then** $x_{i+1} \leftarrow y$; $x_{j-1} \leftarrow x$
13   **else return** *not single-peaked*
14 **if** $L=\{x\}$ **then**
15   **if** $x=x_{i+1}$ **then** Make-axis($C'\setminus\{x\}$,$(x_1,\ldots,x_i,x_{i+1})$,$(x_j,\ldots,x_m)$)
16   **else** Make-axis($C'\setminus\{x\}$,$(x_1,\ldots,x_i)$,$(x_{j-1},x_j,\ldots,x_m)$)
17 Make-axis($C'\setminus\{x,y\}$,$(x_1,\ldots,x_i,x_{i+1})$,$(x_{j-1},x_j,\ldots,x_m)$)

Lemma 1. *$\mathcal{P}$ is single-peaked iff it has no forbidden minor.*

Proof (sketch) *Necessity*: it suffices to check that none of the five forbidden minors is single-peaked, since the single-peakedness property is closed under taking minors.
Sufficiency: run Algorithm 2 and suppose that it returns *not single-peaked*. If it stops at Line 5, then $\mathcal{P}$ has a minor $\mathcal{T}_1$ or $\mathcal{T}_2$. Otherwise it stops at Line 10 or 13 and $\mathcal{P}$ has a minor $\mathcal{F}_1$, $\mathcal{F}_2$ or $\mathcal{F}_3$. ∎

The rest of the section is devoted to the problem of determining an ordered partition of minimum width among the ones that are compatible with clustered single-peakedness. Note that Elkind et al. [7] studied a closely related problem, namely finding an ordered partition $(C_1,\ldots,C_q)$ maximizing q. Both problems are not equivalent, as shown by the following example.

Example 5. *Consider the preference profile matrix $\mathcal{P}$:*

$$\mathcal{P}=\begin{pmatrix} d & d \\ x & a \\ y & v \\ c & b \\ b & c \\ a & x \\ v & y \end{pmatrix}$$

Both partitions $(abcv,d,x,y)$ and $(v,a,d,bcxy)$ maximize q and are compatible with clustered single-peakedness, but (abv,d,cxy) is the only partition that minimizes the single-peaked width.

However, for *narcissistic preferences* [3, 13], one can show that the algorithm proposed by Elkind et al. for their problem returns an ordered partition of minimum width. Nevertheless, our approach proves that there is a unique (up to reversal) ordered partition maximizing q. Preferences are said to be *narcissistic* when each candidate is most preferred by some voter. In politics, as soon as the candidates are voting, this assumption seems reasonable. In the remainder, we prove the following result:

Theorem 2. *Finding the single-peaked width is polynomial if $\mathcal{P}$ is narcissistic.*

For each voter $v\in V$ and candidates $a,b\in C$ we denote $I_v(a,b):=\{c\in C : c=a$ or $c=b$ or $a\succ_v c\succ_v b$ or $b\succ_v c\succ_v a\}$ the set of candidates between a and b in the preferences of voter v.

By convention, $I_v(a,a) = \{a\}$. A subset I of C is called an *interval of* $\mathcal{P}$ if for each $v \in V$, one can choose two candidates $a, b \in I$ such that $I = I_v(a,b)$. This definition coincides with the notion of *clone set* studied by Elkind et al. [7]. Notice that the set of intervals $\mathcal{I}$ of $\mathcal{P}$ is not closed under taking subsets. Nevertheless, it is closed under intersection [7]. Given $a, b \in C$, the minimal interval w.r.t. inclusion that contains a and b is thus uniquely defined: we denote it by $I(a,b)$. The following lemma will prove useful in order to design an algorithm able to compute a partition compatible with clustered single-peakedness. For simplicity, if $\mathcal{P}'$ is isomorphic to $\mathcal{P}$ for ϕ, we write $I(x,y)$ for $I(\phi(x), \phi(y))$.

Lemma 2. *The following properties hold:*
- *If $\mathcal{T}_1$ is a minor of $\mathcal{P}$, then $I(a,b) = I(a,c) = I(b,c)$;*
- *If $\mathcal{T}_2$ is a minor of $\mathcal{P}$, then $I(a,b)$ and $I(a,c)$ include $I(b,c)$;*
- *If $\mathcal{F}_1$ is a minor of $\mathcal{P}$, then $I(a,b)$, $I(a,c)$, $I(a,d)$, $I(b,c)$ and $I(c,d)$ include $I(b,d)$;*
- *If $\mathcal{F}_2$ is a minor of $\mathcal{P}$, then $I(a,b)$, $I(a,c)$, $I(a,d)$, $I(b,d)$ and $I(c,d)$ include $I(b,c)$;*
- *If $\mathcal{F}_3$ is a minor of $\mathcal{P}$, then*
  - *$I(a,c)$, $I(a,d)$, $I(b,d)$ and $I(c,d)$ include $I(a,b)$,*
  - *$I(a,c)$, $I(a,d)$, $I(b,d)$ and $I(c,d)$ include $I(b,c)$.*

Proof (sketch) Let v be the voter of the first column of $\mathcal{T}_1$. Since $b \in I_v(a,c)$, it follows that $b \in I(a,c)$. Thus $I(a,b)$ and $I(b,c) \subseteq I(a,c)$. The second column gives $I(b,c)$ and $I(a,c) \subseteq I(a,b)$, and the third column gives $I(a,c)$ and $I(a,b) \subseteq I(b,c)$. Finally $I(a,b)$ = $I(a,c)$ = $I(b,c)$. The proofs for the four other forbidden minors go along the same lines. ■

We propose a greedy algorithm to compute the clusters of an ordered partition compatible with clustered single-peakedness. This algorithm proceeds by *contracting* candidates so that no forbidden minor remains in the preference profile matrix. *Contracting two candidates* $a, b \in C$ consists in contracting $I(a,b)$. *Contracting an interval* I consists in collapsing all candidates in I into a single "cluster" candidate. This amounts to choosing a representative in I and removing from $\mathcal{P}$ all the other candidates in I. For instance, contracting b and d in $\mathcal{P}$ yields cluster $\{b,d,e\}$ (since $I(b,d) = \{b,d,e\}$) and profile $\mathcal{P}'$:

$$\mathcal{P} = \begin{pmatrix} a & c \\ d & d \\ e & b \\ b & e \\ c & a \end{pmatrix} \qquad \mathcal{P}' = \begin{pmatrix} a & c \\ b & b \\ c & a \end{pmatrix}$$

Notice that the preference profile matrix $\mathcal{P}'$ obtained by contracting an interval of $\mathcal{P}$ is a minor of $\mathcal{P}$. Note also that if $I, J \in \mathcal{I}$ are two intervals of $\mathcal{P}$, then the two minors obtained from $\mathcal{P}$ either by contracting I then J, or by contracting J then I coincide (even if I and J overlap). Besides, if $\mathcal{P}'$ is a minor of $\mathcal{P}$ and $\mathcal{F}'$ is a minor of $\mathcal{P}'$, then $\mathcal{F}'$ is also a minor of $\mathcal{P}$. The greedy procedure is detailed in Algorithm 3. The termination follows from the fact that contracting candidates cannot create new forbidden minors.

Example 6. *Consider the preference profile matrix $\mathcal{P}$ in Figure 2 and apply Algorithm 3, assume that it detects:*

$$\textit{the minor} \begin{pmatrix} g & g & c \\ c & a & a \\ a & c & g \end{pmatrix} \textit{ and then the minor} \begin{pmatrix} f & h \\ g & g \\ e & e \\ h & f \end{pmatrix}$$

Algorithm 3: Greedy algorithm

let $\mathcal{P}'$ be a minor of $\mathcal{P}$ isomorphic for ϕ to:
$\mathcal{T}_1$. Contract $\phi(a)$ and $\phi(b)$
$\mathcal{T}_2$. Contract $\phi(b)$ and $\phi(c)$
$\mathcal{F}_1$. Contract $\phi(b)$ and $\phi(d)$
$\mathcal{F}_2$. Contract $\phi(b)$ and $\phi(c)$
$\mathcal{F}_3$. Contract $\phi(a)$ and $\phi(b)$, or $\phi(b)$ and $\phi(c)$
apply these contractions (non-deterministically) until no forbidden minor remains.

The first minor is isomorphic to $\mathcal{T}_2$ for $\phi(a) = g$, $\phi(b) = a$ and $\phi(c) = c$. Therefore candidates a and c are contracted. The second minor is isomorphic to $\mathcal{F}_1$ for $\phi(a) = f$, $\phi(b) = e$, $\phi(c) = h$ and $\phi(d) = g$. Therefore candidates e and g are contracted. Taking candidate a (resp. e) as the representative of cluster $\{a, c\}$ (resp. $\{e, g\}$), the preference profile becomes:

$$\mathcal{P}' = \begin{pmatrix} f & b & a \\ e & d & f \\ a & h & e \\ b & e & b \\ d & f & d \\ h & a & h \end{pmatrix}$$

There is no more forbidden minor in the preference profile, and thus the greedy procedure stops. The clusters are $\{a, c\}$ and $\{e, g\}$.

This algorithm is polynomial since the forbidden minors can be enumerated in $O(m^3n^3)$ for $\mathcal{T}_1$, $\mathcal{T}_2$, and $O(m^4n^2)$ for $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$. The clusters identified by the algorithm belong to an ordered partition compatible with clustered single-peakedness. The ordered partition itself can be computed by applying Algorithm 2 on the final preference profile. Coming back to Example 6, Algorithm 2 returns axis (h, d, b, e, f, a) on $\mathcal{P}'$, which corresponds to the ordered partition (h, d, b, eg, f, ac) since e (resp. a) is the representative of $\{e, g\}$ (resp. $\{a, c\}$). This is an ordered partition of minimum width for this profile. However, in the general case, the width of the returned ordered partition is not guaranteed to be minimal. We now show how to refine the greedy procedure to get an ordered partition of minimum width when preferences are narcissistic. To this end, we introduce a notion of similarity between candidates that enables us to identify necessary and sufficient contractions for clustered single-peakedness.

Definition 4. *Two candidates a and b are said to be* similar *if they belong to the same cluster in all ordered partitions w.r.t. which $\mathcal{P}$ is clustered single-peaked.*

It results from Lemma 2 that the following properties hold:

- If $\mathcal{T}_1$ (T_2) is a minor of $\mathcal{P}$, then a and b (b and c) are similar;
- If $\mathcal{F}_1$ (F_2) is a minor of $\mathcal{P}$, then b and d (b and c) are similar;
- If $\mathcal{F}_3$ is a minor of $\mathcal{P}$, then
  - if $I(b, c) \subseteq I(a, b)$, then b and c are similar;
  - if $I(a, b) \subseteq I(b, c)$, then a and b are similar.

These properties imply that all contractions but one ($\mathcal{F}_3$) in the greedy algorithm cover candidates which belong to the same cluster in any ordered partition of minimum width. The only case of a forbidden minor that cannot be removed from $\mathcal{P}$ by contracting similar candidates is thus $\mathcal{F}_3$ when $I(a, b)$ and $I(b, c)$ intersect properly, i.e. $I(a, b) \not\subseteq I(b, c)$ and $I(b, c) \not\subseteq I(a, b)$. We call such forbidden minors *ambiguous*. If one finds an ambiguous minor $\mathcal{M}$, at least two candidates in $\mathcal{M}$

must be in the same cluster. Nevertheless the single-peaked width of an ordered partition depends on the choice of the candidates to contract. Furthermore this choice does not only depend on the maximum number of candidates involved in the possible interval contractions. For instance, consider the preference profile matrix $\mathcal{P}$ of Example 5 which has the following minor:

$$\begin{pmatrix} d & d \\ c & v \\ b & b \\ v & c \end{pmatrix}$$

The smallest contraction implied by the given minor would be to contract b and c (2 candidates in the interval). But (abv, d, cxy), where b and c are not in the same cluster, is the only minimum width ordered partition compatible with clustered single-peakedness.

For this reason, the greedy algorithm may fail to provide clusters belonging to an ordered partition of minimum width. However when preferences are narcissistic, no ambiguous minor can exist in the preference profile matrix. Assume indeed that there exists a minor $\mathcal{M}$ isomorphic to $\mathcal{F}_3$ for ϕ. Since $\mathcal{P}$ is narcissistic, candidate $\phi(b)$ is the most preferred one for some voter v, and consequently: $\phi(b) \succ_v \phi(a) \succ_v \phi(c)$ or $\phi(b) \succ_v \phi(c) \succ_v \phi(a)$. Therefore we have $I(a,b) \subseteq I(b,c)$ or $I(b,c) \subseteq I(a,b)$. The minor is thus unambiguous. To obtain an optimal greedy algorithm for narcissistic preferences, contraction related to $\mathcal{F}_3$ must then be modified as follows:

let v be a voter whose most preferred candidate is $\phi(b)$
if $\phi(b) \succ_v \phi(a) \succ_v \phi(c)$ then contract $\phi(a)$ and $\phi(b)$ else contract $\phi(b)$ and $\phi(c)$.

Furthermore, the greedy algorithm uses necessary and sufficient contractions to make the profile clustered single-peaked, and thus partition $(C_1, \dots, C_q)$ of minimum width is clearly unique under maximizing the number q of clusters.

5 Conclusion

An interesting open question is whether there exists a general polynomial algorithm to compute the single-peaked width (not necessarily in the narcissistic case). Adapting the PQ-trees based algorithm of Elkind et al. [7] to our problem could work. Besides, the concept of minors could be a tool for finding a short validity proof.

6 Acknowledgements

We wish to thank the anonymous reviewers of ECAI and COMSOC for their fruitful comments on an earlier version of the paper. This research has been supported by the ANR project ComSoc (ANR-09-BLAN-0305).

References

[1] M. Balinski. On finding integer solutions to linear programs. In *Proc. IBM Scientific Computing Symposium on Combinatorial Problems*, pages 225–248, 1966.

[2] M. A. Ballester and G. Haeringer. A characterization of the single-peaked domain. *Social Choice and Welfare*, 36(2):305–322, 2011.

[3] J. Bartholdi III and M. A. Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4):165 – 169, 1986.

[4] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. Available at SSRN: http://ssrn.com/abstract=1952497.

[5] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[6] J. R. Chamberlin and P. N. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *The American Political Science Review*, 77(3):pp. 718–733, 1983.

[7] E. Elkind, P. Faliszewski, and A. Slinko. Clone structures in voters' preferences. *ArXiv e-prints*, 1110.3939, 2011.

[8] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *Proceedings of 18th European Conference on Artificial Intelligence (ECAI 2008)*, pages 366–370. IOS Press, 2008.

[9] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of manipulative attacks in nearly single-peaked electorates. In *TARK*, pages 228–237, 2011.

[10] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *IJCAI*, pages 280–286, 2011.

[11] R. Potthoff and S. Brams. Proportional representation: broadening the options. *Journal of Theoretical Politics*, 10(2):pp. 147–178, 1998.

[12] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[13] M. A. Trick. Recognizing single-peaked preferences on a tree. *Mathematical Social Sciences*, 17(3):329–334, 1989.

Denis Cornaz
University Paris Dauphine
LAMSADE
Place du Marchal de Lattre de Tassigny
75775 Paris cedex 16, France
Email: `denis.cornaz@dauphine.fr`

Lucie Galand
University Paris Dauphine
LAMSADE
Place du Marchal de Lattre de Tassigny
75775 Paris cedex 16, France
Email: `lucie.galand@dauphine.fr`

Olivier Spanjaard
University of Pierre et Marie Curie
LIP6
4 Place Jussieu
75252 Paris Cedex 05, France
Email: `olivier.spanjaard@lip6.fr`

The Common Structure of Paradoxes in Aggregation Theory

Umberto Grandi

Abstract

In this paper we analyse some of the classical paradoxes in Social Choice Theory (namely, the Condorcet paradox, the discursive dilemma, the Ostrogorski paradox and the multiple election paradox) using a general framework for the study of aggregation problems called binary aggregation with integrity constraints. We provide a definition of paradox that is general enough to account for the four cases mentioned, and identify a common structure in the syntactic properties of the rationality assumptions that lie behind such paradoxes. We generalise this observation by providing a full characterisation of the set of rationality assumptions on which the majority rule *does not* generate a paradox.

1 Introduction

Most work in Social Choice Theory (SCT) started with the observation of paradoxical situations. From the Marquis de Condorcet (1785) to more recent American court cases (Kornhauser and Sager, 1986), a wide collection of paradoxes have been analysed and studied in the literature on Social Choice Theory (see, e.g., Nurmi, 1999). More recently, researchers in Artificial Intelligence (AI) have become interested in the study of collective choice problems in which the set of alternatives has a combinatorial structure (Lang, 2007; Xia et al., 2011). Novel paradoxical situations emerged from the study of these situations, and the combinatorial structure of the domains gave rise to interesting computational challenges.

This paper concentrates on the use of the majority rule on binary combinatorial domains, and investigates the question of what constitutes a paradox in such a setting. We identify a common structure behind the most classical paradoxes in SCT, putting forward a general definition of paradox in aggregation theory. By characterising paradoxical situations by means of computationally recognisable properties, we aim at providing more domain-specific research with new tools for the development of safe procedures for collective decision making.

We base the analysis on our previous work on binary aggregation with integrity constraints (Grandi and Endriss, 2011), which constitutes a general framework for the study of aggregation problems. In this setting, a set of individuals needs to take a decision over a set of binary issues, and these choices are then aggregated into a collective one. Given a rationality assumption that binds the choices of the individuals, we define a paradox as a situation in which all individuals are rational but the collective outcome is not. We present some of the most well-known paradoxes that arise from the use of the majority rule in different contexts, and we show how they can be expressed in binary aggregation as instances of this general definition. Our analysis focuses on the Condorcet paradox (1785), the discursive dilemma in judgment aggregation (List and Pettit, 2002), the Ostrogorski paradox (1902) and the more recent work of Brams et al. (1998) on multiple election paradoxes.

Such a uniform representation of the most important paradoxes in SCT enables us to make a crucial observation concerning the syntactic structure of the rationality assumptions that lie behind these paradoxes. We represent rationality assumptions by means of propositional formulas, and we observe that all formulas formalising a number of classical paradoxes feature a disjunction of literals of size at least 3. This observation can be generalised to a full characterisation of the rationality assumptions on which the majority rule does not generate a paradox, and in Theorem 4 we identify them as those formulas that are equivalent to a conjunction of clauses of size at most 2.

The paper is organised as follows. In Section 2 we give the basic definitions of the framework of binary aggregation with integrity constraints, and we provide a general definition of paradox. In Section 3 we show how a number of paradoxical situations in SCT can be seen as instances of our general definition of paradox, and we identify a syntactic property that is common to all paradoxical rationality assumptions. Section 4 provides a characterisation of the paradoxical situations for the majority rule and Section 5 concludes the paper.

2 Binary Aggregation with Integrity Constraints

In this section we provide the basic definitions of the framework of binary aggregation with integrity constraints (Grandi and Endriss, 2011), based on work by Wilson (1975) and Dokow and Holzman (2010). In this setting, a number individuals each need to make a yes/no choice regarding a number of issues and these choices then need to be aggregated into a collective choice. Paradoxical situations may occur when a set of individual choices that is considered rational leads to a collective outcome which fails to satisfy the same rationality assumption of the individuals.

2.1 Terminology and Notation

Let $\mathcal{I} = \{1, \dots, m\}$ be a finite set of *issues*, and let $\mathcal{D} = D_1 \times \cdots \times D_m$ be a *boolean combinatorial domain*, i.e., $|D_i| = 2$ for all $i \in \mathcal{I}$. Without loss of generality we assume that $D_j = \{0, 1\}$ for all j. Thus, given a set of issues $\mathcal{I}$, the domain associated with it is $\mathcal{D} = \{0, 1\}^{\mathcal{I}}$. A *ballot* B is an element of $\mathcal{D}$.

In many applications it is necessary to specify which elements of the domain are rational and which should not be taken into consideration. Propositional logic provides a suitable formal language to express possible restrictions of rationality on binary combinatorial domains. If $\mathcal{I}$ is a set of m issues, let $PS = \{p_1, \dots, p_m\}$ be a set of propositional symbols, one for each issue, and let $\mathcal{L}_{PS}$ be the propositional language constructed by closing PS under propositional connectives. For any formula $\varphi \in \mathcal{L}_{PS}$, let $\mathrm{Mod}(\varphi)$ be the set of assignments that satisfy φ. For example, $\mathrm{Mod}(p_1 \wedge \neg p_2) = \{(1, 0, 0), (1, 0, 1)\}$ when $PS = \{p_1, p_2, p_3\}$. An *integrity constraint* is any formula $\mathrm{IC} \in \mathcal{L}_{PS}$.

Integrity constraints can be used to define what tuples in $\mathcal{D}$ we consider *rational* choices. Any ballot $B \in \mathcal{D}$ is an assignment to the variables $p_1, \dots, p_m$, and we call B a *rational ballot* if it satisfies the integrity constraint IC, i.e., if B is an element of Mod(IC). In the sequel we shall use the terms "integrity constraints" and "rationality assumptions" interchangeably.

Let $\mathcal{N} = \{1, \dots, n\}$ be a finite set of *individuals*. We make the assumption that there are at least 2 individuals. Each individual submits a ballot $B_i \in \mathcal{D}$ to form a *profile* $\boldsymbol{B} = (B_1, \dots, B_n)$. We write b_j for the jth element of a ballot B, and $b_{i,j}$ for the jth element of ballot B_i within a profile $\boldsymbol{B} = (B_1, \dots, B_n)$. Given a finite set of issues $\mathcal{I}$ and a finite set of individuals $\mathcal{N}$, an *aggregation procedure* is a function $F : \mathcal{D}^{\mathcal{N}} \to \mathcal{D}$, mapping each profile of binary ballots to an element of $\mathcal{D}$. Let $F(\boldsymbol{B})_j$ denote the result of the aggregation of profile $\boldsymbol{B}$ on issue j.

2.2 A General Definition of Paradox

Consider the following example: Let $\mathrm{IC} = p_1 \wedge p_2 \to p_3$ and suppose there are three individuals, choosing ballots $(0, 1, 0)$, $(1, 0, 0)$ and $(1, 1, 1)$. Their choices are rational (they all satisfy IC). However, if we employ the *majority rule*, i.e., we accept an issue j if and only if a majority of individuals do, we obtain the ballot $(1, 1, 0)$ as collective outcome, which fails to be rational. This kind of observation is often referred to as a paradox.

We now give a general definition of paradoxical behaviour of an aggregation procedure in terms of the violation of certain rationality assumptions.

Definition 1. A *paradox* is a triple $(F, \boldsymbol{B}, \text{IC})$, where F is an aggregation procedure, $\boldsymbol{B}$ is a profile in $\mathcal{D}^{\mathcal{N}}$, IC is an integrity constraint in $\mathcal{L}_{PS}$, and $B_i \in \text{Mod(IC)}$ for all $i \in \mathcal{N}$ but $F(\boldsymbol{B}) \notin \text{Mod(IC)}$.

A closely related notion is that of collective rationality:

Definition 2. Given an integrity constraint IC $\in \mathcal{L}_{PS}$, an aggregation procedure F is called *collectively rational* (CR) with respect to IC, if for all rational profiles $\boldsymbol{B} \in \text{Mod(IC)}^{\mathcal{N}}$ we have that $F(\boldsymbol{B}) \in \text{Mod(IC)}$.

Thus, F is CR with respect to IC if it *lifts* the rationality assumption given by IC from the individual to the collective level, i.e., if $F(\boldsymbol{B}) \in \text{Mod(IC)}$ whenever $B_i \in \text{Mod(IC)}$ for all $i \in \mathcal{N}$. An aggregation procedure that is CR with respect to IC cannot generate a paradoxical situation with IC as integrity constraint. Given an aggregation procedure F, let $\mathcal{LF}[F] = \{\varphi \in \mathcal{L}_{PS} \mid F \text{ is CR with respect to } \varphi\}$ be the set of integrity constraints that are lifted by F.

3 Unifying Paradoxes in Binary Aggregation

In this section we present a number of classical paradoxes from SCT, and we show how they can be seen as instances of our Definition 1. In Section 3.1 we introduce the Condorcet paradox, and we show how settings of preference aggregation can be seen as instances of binary aggregation by devising a suitable integrity constraint. Section 3.2 repeats this construction for the framework of judgment aggregation and for the discursive dilemma. In Section 3.3 we then deal with the Ostrogorski paradox, in which a paradoxical feature of representative majoritarian systems is analysed, and in Section 3.4 we focus on the paradox of multiple elections. In Section 3.5 we conclude by identifying a common structure in the integrity constraints that lie behind those paradoxes.

3.1 The Condorcet Paradox and Preference Aggregation

One of the earliest observation of paradoxical behaviour of the majority rule was made by the Marquis de Condorcet in 1785. A simple version of the paradox he discovered is explained in the following paragraphs:

> **Condorcet Paradox**. Three individuals need to decide on the ranking of three alternatives $\{\triangle, \bigcirc, \square\}$. Each individual expresses her own ranking and the collective outcome is aggregated by pairwise majority: an alternative is preferred to a second one if and only if a majority of the individuals prefer the first alternative to the second. Consider the following situation:
>
> $$\begin{array}{c} \triangle <_1 \bigcirc <_1 \square \\ \square <_2 \triangle <_2 \bigcirc \\ \bigcirc <_3 \square <_3 \triangle \\ \hline \triangle < \bigcirc < \square < \triangle \end{array}$$
>
> When computing the outcome of the pairwise majority rule, we notice that there is a majority of individuals preferring the circle to the triangle ($\triangle < \bigcirc$); that there is a majority of individuals preferring the square to the circle ($\bigcirc < \square$); and, finally, that there is a majority of individuals preferring the triangle to the square ($\square < \triangle$). The resulting outcome fails to be a linear order, giving rise to a circular collective preference between the alternatives.

3.1.1 Preference Aggregation

Condorcet's paradox was rediscovered in the second half of the XXth century while a whole theory of *preference aggregation* was being developed (see, e.g., Gaertner, 2006). This framework considers a finite set of individuals $\mathcal{N}$ expressing preferences over a finite set of alternatives $\mathcal{X}$. A preference relation is represented by a binary relation over $\mathcal{X}$. Preference relations are traditionally assumed to be *weak orders*, i.e., reflexive, transitive and complete binary relations. Another common assumption is representing preferences as *linear orders*, i.e., irreflexive, transitive and complete binary relations. In the sequel we shall assume that preferences are represented as linear orders, writing aPb for "alternative a is strictly preferred to b". Each individual in $\mathcal{N}$ submits a linear order P_i, forming a profile $\boldsymbol{P} = (P_1, \ldots, P_{|\mathcal{N}|})$. Let $\mathcal{L}(\mathcal{X})$ denote the set of all linear orders on $\mathcal{X}$. Given a finite set of individuals $\mathcal{N}$ and a finite set of alternatives $\mathcal{X}$, a *social welfare function* is a function $F : \mathcal{L}(\mathcal{X})^{\mathcal{N}} \to \mathcal{L}(X)$.

3.1.2 Translation

Given a preference aggregation problem defined by a set of individuals $\mathcal{N}$ and a set of alternatives $\mathcal{X}$, let us consider the following setting for binary aggregation. Define a set of issues $\mathcal{I}_{\mathcal{X}}$ as the set of all pairs (a, b) in $\mathcal{X}$. The domain $\mathcal{D}_{\mathcal{X}}$ of aggregation is $\{0,1\}^{|\mathcal{X}|^2}$. In this setting, a binary ballot B corresponds to a binary relation P over $\mathcal{X}$: $B_{(a,b)} = 1$ if and only if a is in relation to b (aPb). Given this representation, we can associate with every SWF for $\mathcal{X}$ and $\mathcal{N}$ an aggregation procedure that is defined on a subdomain of $\mathcal{D}_{\mathcal{X}}^{\mathcal{N}}$. We now characterise this domain as the set of models of a suitable integrity constraint.

Using the propositional language $\mathcal{L}_{PS}$ constructed over the set $\mathcal{I}_{\mathcal{X}}$, we can express properties of binary ballots in $\mathcal{D}_{\mathcal{X}}$. In this case $\mathcal{L}_{PS}$ consists of $|\mathcal{X}|^2$ propositional symbols, which we call p_{ab} for every issue (a, b). The properties of linear orders can be enforced on binary ballots using the following set of integrity constraints, which we shall call $\text{IC}_<$:[1]

Irreflexivity: $\neg p_{aa}$ for all $a \in \mathcal{X}$
Completeness: $p_{ab} \vee p_{ba}$ for all $a \neq b \in \mathcal{X}$
Transitivity: $p_{ab} \wedge p_{bc} \rightarrow p_{ac}$ for $a, b, c \in \mathcal{X}$ pairwise distinct

Note that the size of this set of integrity constraints is polynomial in the number of alternatives in $\mathcal{X}$. In case preferences are expressed using weak orders rather than linear orders, it is sufficient to replace the integrity constraints of irreflexivity in $\text{IC}_<$ with their negation to obtain a similar correspondence between SWFs and aggregation procedures.

3.1.3 The Condorcet Paradox in Binary Aggregation

The translation presented in the previous section enables us to express the Condorcet paradox in terms of Definition 1. Let $\mathcal{X} = \{\triangle, \bigcirc, \square\}$ and let $\mathcal{N}$ contain three individuals. Consider the profile $\boldsymbol{B}$ for $\mathcal{I}_{\mathcal{X}}$ in the following table, where we have omitted the values of the reflexive issues $(\triangle, \triangle)$ (always 0 by $\text{IC}_<$), and specified the value of only one of $(\triangle, \bigcirc)$ and $(\bigcirc, \triangle)$ (the other can be obtained by taking the opposite of the value of the first), and accordingly for the other alternatives.

	$\triangle\bigcirc$	$\bigcirc\square$	$\triangle\square$
Agent 1	1	1	1
Agent 2	1	0	0
Agent 3	0	1	0
Maj	1	1	0

[1] We will use the notation IC both for a single integrity constraint and for a set of formulas—in the latter case considering as the actual constraint the conjunction of all the formulas in IC.

Every individual ballot satisfies $IC_<$, but the outcome obtained using the majority rule *Maj* (which corresponds to pairwise majority in preference aggregation) does not satisfy $IC_<$: the formula $p_{\triangle\bigcirc} \wedge p_{\bigcirc\square} \rightarrow p_{\triangle\square}$ is falsified by the outcome. Therefore, $(Maj, \boldsymbol{B}, IC_<)$ is a paradox by Definition 1.

3.2 The Discursive Dilemma and Judgment Aggregation

The discursive dilemma emerged from the formal study of court cases that was carried out in recent years in the literature on law and economics, generalising the observation of a paradoxical situation known as the "doctrinal paradox" (Kornhauser and Sager, 1986). Such a setting was first given mathematical treatment by List and Pettit (2002), giving rise to an entirely new research area in SCT known as *judgment aggregation*.

> **Discursive Dilemma**. A court of three judges has to decide on the liability of a defendant under the charge of breach of contract. An individual is considered liable if there was a valid contract and her behaviour was such as to be considered a breach of the contract. The court takes three majority decisions on the following issues: there was a valid contract (α), the individual broke the contract (β), the defendant is liable ($\alpha \wedge \beta$). Consider the following situation:

	α	β	$\alpha \wedge \beta$
Judge 1	yes	yes	yes
Judge 2	no	yes	no
Judge 3	yes	no	no
Majority	yes	yes	no

> All judges express consistent judgments: they accept the third proposition if and only if the first two are accepted. However, even if there is a majority of judges who believe that there was a valid contract, and even if there is a majority of judges who believe that the individual broke the contract, the individual is considered *not liable* by a majority of the individuals.

3.2.1 Judgment Aggregation

Judgement aggregation (JA) considers problems in which a finite set of individuals $\mathcal{N}$ has to generate a collective judgment over a set of interconnected propositional formulas (see, e.g., List and Puppe, 2009). Formally, given a propositional language $\mathcal{L}$, an *agenda* is a finite nonempty subset $\Phi \subseteq \mathcal{L}$ that does not contain doubly-negated formulas and is closed under complementation (i.e, $\alpha \in \Phi$ whenever $\neg\alpha \in \Phi$, and $\neg\alpha \in \Phi$ for non-negated $\alpha \in \Phi$).

Each individual in $\mathcal{N}$ expresses a *judgment set* $J \subseteq \Phi$, as the set of those formulas in the agenda that she judges to be true. Every individual judgment set J is assumed to be *complete* (i.e., for each $\alpha \in \Phi$ either α or its complement are in J) and *consistent* (i.e., there exists an assignment that makes all formulas in J true). Denote by $\mathcal{J}(\Phi)$ the set of all complete and consistent subsets of Φ. Given a finite agenda Φ and a finite set of individuals $\mathcal{N}$, a *JA procedure* for Φ and $\mathcal{N}$ is a function $F : \mathcal{J}(\Phi)^{\mathcal{N}} \rightarrow 2^{\Phi}$.

3.2.2 Translation

Given a JA framework defined by an agenda Φ and a set of individuals $\mathcal{N}$, let us now construct a setting for binary aggregation with integrity constraints that interprets it. Let the set of issues $\mathcal{I}_\Phi$ be equal to the set of formulas in Φ. The domain $\mathcal{D}_\Phi$ of aggregation is therefore $\{0,1\}^{|\Phi|}$. In this setting, a binary ballot B corresponds to a judgment set: $B_\alpha = 1$ if and only if $\alpha \in J$. Given this representation, we can associate with every JA procedure for Φ and $\mathcal{N}$ a binary aggregation procedure on a subdomain of $\mathcal{D}_\Phi^{\mathcal{N}}$.

As we did for the case of preference aggregation, we now define a set of integrity constraints for $\mathcal{D}_\Phi$ to enforce the properties of consistency and completeness of individual judgment sets. Recall that the propositional language is constructed in this case on $|\Phi|$ propositional symbols p_α, one for every $\alpha \in \Phi$. Call an inconsistent set of formulas each proper subset of which is consistent *minimally inconsistent set* (mi-set). Let IC_Φ be the following set of integrity constraints:

Completeness: $p_\alpha \vee p_{\neg\alpha}$ for all $\alpha \in \Phi$
Consistency: $\neg(\bigwedge_{\alpha \in S} p_\alpha)$ for every mi-set $S \subseteq \Phi$

While the interpretation of the first formula is straightforward, we provide some further explanation for the second one. If a judgment set J is inconsistent, then it contains a minimally inconsistent set, obtained by sequentially deleting one formula at the time from J until it becomes consistent. This implies that the constraint previously introduced is falsified by the binary ballot that represents J, as all issues associated with formulas in a mi-set are accepted. *Vice versa*, if all formulas in a mi-set are accepted by a given binary ballot, then clearly the judgment set associated with it is inconsistent.

Note that the size of IC_Φ might be exponential in the size of the agenda. This is in agreement with considerations of computational complexity: Since checking the consistency of a judgment set is NP-hard, while model checking on binary ballots is polynomial, the translation from JA to binary aggregation must contain a superpolynomial step (unless P=NP).

3.2.3 The Discursive Dilemma in Binary Aggregation

The same procedure that we have used to show that the Condorcet paradox is an instance of our general definition of paradox applies here for the case of the discursive dilemma. Let Φ be the agenda $\{\alpha, \beta, \alpha \wedge \beta\}$, in which we have omitted negated formulas, as for any $J \in \mathcal{J}(\Phi)$ their acceptance can be inferred from the acceptance of their positive counterparts. Consider the profile $\boldsymbol{B}$ for $\mathcal{I}_\Phi$ described in the following profile:

	α	β	$\alpha \wedge \beta$
Judge 1	1	1	1
Judge 2	0	1	0
Judge 3	1	0	0
Maj	1	1	0

Every individual ballot satisfies IC_Φ, while the outcome obtained by using the majority rule contradicts one of the constraints of consistency, namely $\neg(p_\alpha \wedge p_\beta \wedge p_{\neg(\alpha\wedge\beta)})$. Hence, $(Maj, \boldsymbol{B}, \text{IC}_\Phi)$ constitutes a paradox by Definition 1.

3.3 The Ostrogorski Paradox

A less well-known paradox concerning the use of the majority rule on multiple issues is the Ostrogorski paradox (Ostrogorski, 1902).

> **Ostrogorski Paradox**. Consider the following situation: there is a two party contest between the Mountain Party (MP) and the Plain Party (PP); three individuals (or, equivalently, three equally big groups in an electorate) will vote for one of the two parties if their view agrees with that party on a majority of the three following issues: economic policy (E), social policy (S), and foreign affairs policy (F). Consider the situation described in Table 1. The result of the two party contest, assuming that the party that has the support of a majority of the voters wins, declares the Plain Party the winner. However, a majority of individuals support the Mountain Party both on the economic policy E *and* on

	E	S	F	Party supported
Voter 1	MP	PP	PP	PP
Voter 2	PP	PP	MP	PP
Voter 3	MP	PP	MP	MP
Maj	MP	PP	MP	PP

Table 1: The Ostrogorski paradox.

the foreign policy F. Thus, the elected party (the PP) is in disagreement with a majority of the individuals on a majority of the issues.

3.3.1 The Ostrogorski Paradox in Binary Aggregation

In this section, we provide a binary aggregation setting that represents the Ostrogorski paradox as a failure of collective rationality with respect to a suitable integrity constraint.

Let $\{E, S, F\}$ be the set of issues at stake, and let the set of issues $\mathcal{I}_O = \{E, S, F, A\}$ consist of the same issues plus an extra issue A to encode the support for the first party (MP). A binary ballot over these issues represents the individual view on the three issues E, S and F: if, for instance, $b_E = 1$, then the individual supports the first party MP on the first issue E. Moreover, it also represents the overall support for party MP (in case issue A is accepted) or PP (in case A is rejected). In the Ostrogorski paradox, an individual votes for a party if and only if she agrees with that party on a majority of the issues. This rule can be represented as a rationality assumption by means of the following integrity constraint IC_O:

$$p_A \leftrightarrow [(p_E \wedge p_S) \vee (p_E \wedge p_F) \vee (p_S \wedge p_F)]$$

An instance of the Ostrogorski paradox can therefore be represented in the following profile:

	E	S	F	A
Voter 1	1	0	0	0
Voter 2	0	0	1	0
Voter 3	1	0	1	1
Maj	1	0	1	0

Each individual accepts issue A if and only if she accepts a majority of the other issues. However, the outcome of the majority rule is a rejection of issue A, even if a majority of the issues gets accepted by the same rule. Therefore, the triple $(Maj, \boldsymbol{B}, \text{IC}_O)$ constitutes a paradox by Definition 1.

3.4 The Paradox of Multiple Elections

Whilst the Ostrogorski paradox was devised to stage an attack against representative systems of collective choice based on parties, the paradox of multiple elections (MEP) is based on the observation that when voting directly on multiple issues, a combination that was not supported nor liked by any of the voters can be the winner of the election (Brams et al., 1998; Lacy and Niou, 2000). While the original model takes into account the full preferences of individuals over combinations of issues, if we focus on only those ballots that are submitted by the individuals, then an instance of the MEP can be represented as a paradox of collective rationality. Let us consider the following simple example:

Multiple election paradox. Suppose three voters need to take a decision over three binary issues A, B and C. Their ballots are described as follows.

	A	B	C
Voter 1	1	0	1
Voter 2	0	1	1
Voter 3	1	1	0
Maj	1	1	1

The outcome of the majority rule is the acceptance of all three issues, even if this combination was not voted for by any of the individuals.

While there seems to be no integrity constraint directly causing this paradox, we may represent the profile in the example above as a situation in which the three individual ballots are bound by a constraint, e.g., $\neg(p_A \wedge p_B \wedge p_C)$. Even if each individual accepts at most two issues, the result of the aggregation is the unfeasible acceptance of all three issues.

As can be deduced from our previous discussion, every instance of the MEP gives rise to several instances of a binary aggregation paradox for Definition 1. To see this, it is sufficient to find an integrity constraint that is satisfied by all individuals and not by the outcome of the aggregation.[2] On the other hand, every instance of Definition 1 in binary aggregation may represent an instance of the MEP, as the irrational outcome cannot have been voted for by any of the individuals.

3.5 The Common Structure of Paradoxical Integrity Constraints

We can now make a crucial observation concerning the syntactic structure of the integrity constraints that formalise the paradoxes we have presented so far.

First, for the case of the Condorcet paradox, we observe that the formula encoding the transitivity of a preference relation is the implication $p_{ab} \wedge p_{bc} \to p_{ac}$. This formula is equivalent to $\neg p_{ab} \vee \neg p_{bc} \vee p_{ac}$, which is a clause of size 3, i.e., it is a disjunction of three different literals. Second, the formula which appears in the translation of the discursive dilemma is also equivalent to a clause of size 3, namely $\neg p_\alpha \vee \neg p_\beta \vee \neg p_{\neg(\alpha\wedge\beta)}$. Third, the formula which formalises the majoritarian constraint underlying the Ostrogorski paradox, is equivalent to the following conjunction of clauses of size 3:

$$(p_A \vee \neg p_E \vee \neg p_F) \wedge (p_A \vee \neg p_E \vee \neg p_S) \wedge (p_A \vee \neg p_S \vee \neg p_F) \wedge \\ \wedge(\neg p_A \vee p_E \vee p_F) \wedge (\neg p_A \vee p_E \vee p_S) \wedge (\neg p_A \vee p_S \vee p_F)$$

Finally, the formula which exemplifies the MEP is equivalent to a negative clause of size 3.

Thus, we observe that the **integrity constraints formalising the most classical paradoxes in aggregation theory all feature a clause of size at least 3.**[3]

4 The Majority Rule: Characterisation of Paradoxes

In this section we generalise the observation made in the previous section, and we provide a full characterisation of the class of integrity constraints that are lifted by the majority rule as those formulas that can be expressed as a conjunction of clauses of maximal size 2.

Under the majority rule, an issue is accepted if and only if a majority of the individuals accept it. Let $N_j^{\boldsymbol{B}}$ be the set of individuals that accept issue j in profile $\boldsymbol{B}$. In case the number of individuals is odd, the *majority rule* (*Maj*) has a unique definition by accepting issue j if and only if $|N_j^{\boldsymbol{B}}| \geqslant \frac{n+1}{2}$. The case of an even number of individuals is more problematic, to account for profiles in which exactly half of the individuals accept an issue

[2] Such a formula always exists: consider the disjunction of the formulas specifying the individual ballots.

[3] This observation is strongly related to a result proven by Nehring and Puppe (2007) in the framework of judgment aggregation, which characterises the set of paradoxical agendas for the majority rule as those agendas containing a minimal inconsistent subset of size at least 3.

and exactly half reject it. We give two different definitions. The *weak majority rule* (*W-Maj*) accepts an issue if and only if $|N_j^{\boldsymbol{B}}| \geqslant \frac{n}{2}$, favouring acceptance. The *strict majority rule* (*S-Maj*) accepts an issue if and only if $|N_j^{\boldsymbol{B}}| \geqslant \frac{n+2}{2}$, favouring rejection.

4.1 Odd Number of Individuals: The Majority Rule

We begin with a base-line result that proves collective rationality of the majority rule in case the integrity constraint is equivalent to a conjunction of 2-clauses. Let *2-clauses* denote the set of propositional formulas in $\mathcal{L}_{PS}$ that are equivalent to a conjunction of clauses of maximal size 2.[4]

Proposition 1. *The majority rule is collectively rational with respect to 2-clauses.*

Proof. Let us first consider the case of a single 2-clause $\text{IC} = \ell_j \vee \ell_k$, where ℓ_j and ℓ_k are two literals, i.e., atoms or negated atoms. A paradoxical profile for the majority rule with respect to this integrity constraint features a first majority of individuals not satisfying literal ℓ_j, and a second majority of individuals not satisfying literal ℓ_k. By the pigeonhole principle these two majorities must have a non-empty intersection, i.e., there exists one individual that does not satisfy both literals ℓ_j and ℓ_k, but this is incompatible with the requirement that all individual ballots satisfy IC. To conclude the proof, it is sufficient to observe that if IC is equivalent to a conjunction of two clauses, then all individuals satisfy each of these clauses, and by the previous discussion all these clauses will also be satisfied by the outcome of the majority rule. □

An easy corollary of this proposition covers the case of just 2 issues:

Corollary 2. *If* $|\mathcal{I}| \leqslant 2$*, then the majority rule is collectively rational with respect to all integrity constraints* $\text{IC} \in \mathcal{L}_{PS}$.

Proof. This follows immediately from Proposition 1 and from the observation that every formula built with two propositional symbols is equivalent to a conjunction of clauses of size at most 2 (e.g., its conjunctive normal form). □

As we have remarked in Section 3.5, all classical paradoxes involving the majority rule can be formalised in our framework by means of an integrity constraint that consists of (or is equivalent to) one or more clauses with size bigger than two. We now generalise this observation to a theorem that completes the characterisation of the integrity constraints lifted by the majority rule. We need some preliminary definitions and a lemma.

Call a *minimally falsifying partial assignment* (mifap-assignment) for an integrity constraint IC an assignment to some of the propositional variables that cannot be extended to a satisfying assignment, although each of its proper subsets can. We first prove a a crucial lemma about mifap-assignments. Given a propositional formula φ, associate with each mifap-assignment ρ for φ a conjunction $C_\rho = \ell_1 \wedge \cdots \wedge \ell_k$, where $\ell_i = p_i$ if $\rho(p_i) = 1$ and $\ell_i = \neg p_i$ if $\rho(p_i) = 0$ for all propositional symbols p_i on which ρ is defined. The conjunction C_ρ represents the mifap-assignment ρ and it is clearly inconsistent with φ.

Lemma 3. *Every non-tautological formula* φ *is equivalent to* $(\bigwedge_\rho \neg C_\rho)$ *with* ρ *ranging over all mifap-assignments of* φ.[5]

Proof. Let A be a total assignment for φ. Suppose $A \not\models \varphi$, i.e., A is a falsifying assignment for φ. Since φ is not a tautology there exists at least one such A. By sequentially deleting propositional symbols from the domain of A we eventually find a mifap-assignment ρ_A for

[4]The set of *2-clauses* can be equivalently defined by closing the set of 2-CNF under logical equivalence.

[5]Formulas $\neg C_\rho$ associated to mifap-assignments ρ for IC are also known as the *prime implicates* of IC. Lemma 3 is a reformulation of the fact that a formula is equivalent to the conjunction of its prime implicates.

φ included in A. Hence, A falsifies the conjunct associated with ρ_A, and thus the whole formula $(\bigwedge_\rho \neg C_\rho)$. Assume now $A \models \varphi$ but $A \not\models (\bigwedge_\rho \neg C_\rho)$. Then there exists a ρ such that $A \models C_\rho$. This implies that $\rho \subseteq A$, as C_ρ is a conjunction. Since ρ is a mifap-assignment for φ, i.e., it is a falsifying assignment for φ, this contradicts the assumption that $A \models \varphi$. □

We are now able to provide a full characterisation of the set of integrity constraints that are lifted by the majority rule in case the set of individuals is odd. Recall from Section 2 that $\mathcal{LF}[F]$ is the set of integrity constraints that are lifted by F.

Theorem 4. *$\mathcal{LF}[Maj]$ = 2-clauses.*[6]

Proof. One direction is entailed by Proposition 1: the majority rule is CR with respect to formulas in *2-clauses*. For the opposite direction assume that IC $\notin$ *2-clauses*, i.e., IC is not equivalent to a conjunction of 2-clauses. We now build a paradoxical profile for the majority rule. By Lemma 3 we know that IC is equivalent to the conjunction $\bigwedge_\rho \neg C_\rho$ of all mifap-assignments ρ for IC. We can therefore infer that at least one mifap-assignment ρ^* has size > 2, for otherwise IC would be equivalent to a conjunction of 2-clauses.

Consider the following profile. Let y_1, y_2, y_3 be three propositional variables that are fixed by ρ^*. Let the first individual i_1 accept the issue associated with y_1 if $\rho(y_1) = 0$, and reject it otherwise, i.e., let $b_{1,1} = 1 - \rho^*(y_1)$. Furthermore, let i_1 agree with ρ^* on the remaining propositional variables. By minimality of ρ^*, this partial assignment can be extended to a satisfying assignment for IC, and let B_{i_1} be such an assignment. Repeat the same construction for individual i_2, this time changing the value of ρ^* on y_2 and extending it to a satisfying assignment to obtain B_{i_2}. The same construction for i_3, changing the value of ρ^* on issue y_3 and extending it to a satisfying assignment B_{i_3}. Recall that there are at least 3 individuals in $\mathcal{N}$. If there are other individuals, let individuals i_{3s+1} have the same ballot B_{i_1}, individuals i_{3s+2} ballot B_{i_2} and individuals i_{3s+3} ballot B_{i_3}. The basic profile for 3 issues and 3 individuals is shown in the following table:

	y_1	y_2	y_3
i_1	1-$\rho^*(y_1)$	$\rho^*(y_2)$	$\rho^*(y_3)$
i_2	$\rho^*(y_1)$	1-$\rho^*(y_2)$	$\rho^*(y_3)$
i_3	$\rho^*(y_1)$	$\rho^*(y_2)$	1-$\rho^*(y_3)$
Maj	$\rho^*(y_1)$	$\rho^*(y_2)$	$\rho^*(y_3)$

As can be seen in the previous table, and easily generalised to the case of more than 3 individuals, there is a majority supporting ρ^* on every variable on which ρ^* is defined. Since ρ^* is a mifap-assignment and therefore cannot be extended to an assignment satisfying IC, the majority rule in this profile is not collectively rational with respect to IC. □

4.2 Even Number of Individuals: Weak and Strict Majority

While a result analogous to Theorem 4 for the case of an even number of individuals cannot be proven, we provide the following result (proof is omitted for lack of space).

Proposition 5. *W-Maj and S-Maj are CR with respect to 2-clauses in which one literal is negative and one is positive. W-Maj is CR with respect to positive 2-clauses, in which all literals occur positively. S-Maj is CR with respect to negative 2-clauses, in which all literals occur negatively.*

[6] This result may be considered a "syntactic counterpart" of a result by Nehring and Puppe (2007) in the framework of judgment aggregation, characterising profiles on which the majority rule outputs a consistent outcome. In the interest of space, we refer to our previous work (Grandi and Endriss, 2011) for a more detailed discussion of the relation between the two results.

5 Conclusions

The first conclusion that can be drawn from this paper dedicated to paradoxes of aggregation is that the majority rule is to be avoided when dealing with collective choices over multiple issues. This fact stands out as a counterpart to May's Theorem (1952), which proves that the majority rule is the only aggregation rule *for a single binary issue* that satisfies a set of highly desirable conditions. The sequence of paradoxes we have analysed in this paper shows that this is not the case when multiple issues are involved. While this fact may not add anything substantially new to the existing literature, the wide variety of paradoxical situations encountered in this paper stresses even further the negative features of the majority rule on multi-issue domains.

A second conclusion is that most paradoxes of SCT share a common structure, and that this structure is formalised by our Definition 1, which stands out as a truly general definition of paradox in aggregation theory. Moreover, by analysing the integrity constraints that underlie some of the most classical paradoxes, we were able to identify a common syntactic feature of paradoxical constraints. Starting from this observation, we have provided a full characterisation of the integrity constraints that are lifted by the majority rule, as those formulas that are equivalent to a conjunction of clauses of size at most 2.

The paradoxical situations presented in this paper constitute a fragment of the problems that can be encountered in the formalisation of collective choice problems. For instance, paradoxical situations concerning voting procedures (Nurmi, 1999), which take as input a set of preferences and output a set of winning candidates, are not included in our analysis.

Recent work on paradoxes of aggregation also pointed at similarities within different frameworks, e.g., comparing the Ostrogorski paradox with the discursive dilemma (Pigozzi, 2005), or proposing a geometric approach for the study of paradoxical situations (Eckert and Klamler, 2009). The MEP gives rise to a different problem than that of collective rationality, not being directly linked to an integrity constraint established in advance. The problem formalised by the MEP is rather the *compatibility* of the outcome of aggregation with the individual ballots. Individuals in such a situation may be forced to adhere to a collective choice which, despite it being rational, they do not perceive as representing their views (Grandi and Pigozzi, 2012). Some answers to the problem raised by the multiple election paradox have already been proposed in the literature on AI, by for instance devising a suitable sequence of local elections (Xia et al., 2011), or by approximating the collective outcome (Conitzer and Xia, 2012).

Elections over multi-issue domains cannot be escaped: not only do they represent a model for the aggregation of more complex objects like preferences and judgments, but they also stand out as one of the biggest challenges to the design of more complex automated systems for collective decision making. A crucial problem in the modelling of real-world situations of collective choice is that of identifying the set of issues that best represent a given domain of aggregation, and devising an integrity constraint that models correctly the correlations between those issues. This problem obviously represents a serious obstacle to a mechanism designer, and is moreover open to manipulation. However, a promising direction for future work consists in structuring collective decision problems with more detailed models *before* the aggregation takes place, e.g., by discovering a shared order of preferential dependencies between issues (Lang and Xia, 2009; Airiau et al., 2011), facilitating the definition of collective choice procedures on complex domains without having to elicit the full preferences of individuals. Such models can be employed in the design and the implementation of automated decision systems, in which a safe aggregation, i.e., one that avoids paradoxical situations, is of the utmost necessity.

References

S. Airiau, U. Endriss, U. Grandi, D. Porello, and J. Uckelman. Aggregating dependency graphs into voting agendas in multi-issue elections. In *Proceedings of IJCAI-2011*, 2011.

S. J. Brams, D. M. Kilgour, and W. S. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15(2):211–236, 1998.

Condorcet, Marie Jean Antoine Nicolas de Caritat, Marquis de. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.* Paris, 1785.

V. Conitzer and L. Xia. Paradoxes of multiple elections: An approximation approach. In *Proceedings of KR-2012*, 2012.

E. Dokow and R. Holzman. Aggregation of binary evaluations. *Journal of Economic Theory*, 145 (2):495–511, 2010.

D. Eckert and C. Klamler. A geometric approach to paradoxes of majority voting in abstract aggregation theory. In *Proceedings of ADT-2009*, 2009.

W. Gaertner. *A Primer in Social Choice Theory.* Oxford University Press, 2006.

U. Grandi and U. Endriss. Binary aggregation with integrity constraints. In *Proceedings IJCAI-2011*, 2011.

U. Grandi and G. Pigozzi. On compatible multi-issue group decisions. In *Proceedings of LOFT-2012*, 2012.

L. A. Kornhauser and L. G. Sager. Unpacking the court. *Yale Law Journal*, 96(1):82–117, 1986.

D. Lacy and E. M. S. Niou. A problem with referendums. *Journal of Theoretical Politics*, 12(1): 5–31, 2000.

J. Lang. Vote and aggregation in combinatorial domains with structured preferences. In *Proceedings of IJCAI-2007*, 2007.

J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57:304–324, 2009.

C. List and P. Pettit. Aggregating sets of judgments: An impossibility result. *Economics and Philosophy*, 18(1):89–110, 2002.

C. List and C. Puppe. Judgment aggregation: A survey. In *Handbook of Rational and Social Choice.* Oxford University Press, 2009.

K. O. May. A set of independent necessary and sufficient conditions for simple majority decision. *Econometrica*, 20(4):680–684, 1952.

K. Nehring and C. Puppe. The structure of strategy-proof social choice. Pt. I: General characterization and possibility results on median spaces. *J. Economic Theory*, 135(1):269–305, 2007.

H. Nurmi. *Voting paradoxes and how to deal with them.* Springer, 1999.

M. Ostrogorski. *La démocratie et l'organisation de partis politiques.* Calmann-Levy, Paris, 1902.

G. Pigozzi. Two aggregation paradoxes in social decision making: the Ostrogorski paradox and the discursive dilemma. *Episteme: A Journal of Social Epistemology*, 2(2):33–42, 2005.

R. B. Wilson. On the theory of aggregation. *Journal of Economic Theory*, 10(1):89–99, 1975.

L. Xia, V. Conitzer, and J. Lang. Strategic sequential voting in multi-issue domains and multiple-election paradoxes. In *Proceedings of ACM-2011*, 2011.

Umberto Grandi
Institute for Logic, Language and Computation
University of Amsterdam
Postbus 94242, 1090 GE Amsterdam, The Netherlands
Email: umberto.uni@gmail.com

The Complexity of Online Manipulation of Sequential Elections

Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe

Abstract

Most work on manipulation assumes that all preferences are known to the manipulators. However, in many settings elections are open and sequential, and manipulators may know the already cast votes but may not know the future votes. We introduce a framework, in which manipulators can see the past votes but not the future ones, to model online coalitional manipulation of sequential elections, and we show that in this setting manipulation can be extremely complex even for election systems with simple winner problems. Yet we also show that for some of the most important election systems such manipulation is simple in certain settings. This suggests that when using sequential voting, one should pay great attention to the details of the setting in choosing one's voting rule.

Among the highlights of our classifications are: We show that, depending on the size of the manipulative coalition, the online manipulation problem can be complete for each level of the polynomial hierarchy or even for PSPACE. And we obtain the most dramatic contrast to date between the nonunique-winner and unique-winner models: Online weighted manipulation for plurality is in P in the nonunique-winner model, yet is coNP-hard (constructive case) and NP-hard (destructive case) in the unique-winner model.

1 Introduction

Voting is a widely used method for preference aggregation and decision-making. In particular, *strategic* voting (or *manipulation*) has been studied intensely in social choice theory (starting with the celebrated work of Gibbard [Gib73] and Satterthwaite [Sat75]) and, in the rapidly emerging area of *computational* social choice, also with respect to its algorithmic properties and computational complexity (starting with the seminal work of Bartholdi, Tovey, and Trick [BTT89]; see the recent surveys by Faliszewski et al. [FP10, FHH10, FHHR09]). This computational aspect is particularly important in light of the many applications of voting in computer science, ranging from meta-search heuristics for the internet [DKNS01], to recommender systems [GMHS99] and multiagent systems in artificial intelligence (see the survey by Conitzer [Con10]).

Most of the previous work on manipulation, however, is concerned with voting where the manipulators know the nonmanipulative votes. Far less attention has been paid (see the related work below) to manipulation in the midst of elections that are modeled as dynamic processes.

We introduce a novel framework for online manipulation, where voters vote in sequence and the current manipulator, who knows the previous votes and which voters are still to come but does not know their votes, must decide—right at that moment—what the "best" vote to cast is. So, while other approaches to sequential voting are game-theoretic, stochastic, or axiomatic in nature (again, see the related work), our approach to manipulation of sequential voting is shaped by the area of "online algorithms" [BE98], in the technical sense of a setting in which one (for us, each manipulative voter) is being asked to make a manipulation decision just on the basis of the information one has in one's hands at the moment even though additional information/system evolution may well be happening down the line. In this area, there are different frameworks for evaluation. But the most attractive one, which pervades the area as a general theme, is the idea that one may want to "maxi-min" things—*one may want to take the action that maximizes the goodness of the set of outcomes that one can expect regardless of what happens down the line from one time-wise*. For example, if the current manipulator's preferences are Alice $>$ Ted $>$ Carol $>$ Bob and if she can cast a (perhaps insincere) vote that ensures that Alice or Ted will be a winner no matter what later voters do, and there is no vote she can cast that ensures that Alice will always be a winner, this maxi-min approach would say that that vote is a "best" vote to cast.

It will perhaps be a bit surprising to those familiar with online algorithms and competitive analysis that in our model of online manipulation we will not use a (competitive) *ratio*. The reason is that voting commonly uses an *ordinal* preference model, in which preferences are total orders of the candidates. It would be a severely improper step to jump from that to assumptions about intensity of preferences and utility, e.g., to assuming that everyone likes her *n*th-to-least favorite candidate exactly *n* times more than she likes her least favorite candidate.

Related Work. Conitzer and Xia [XC10a] (see also the related paper by Desmedt and Elkind [DE10]) define and study the Stackelberg voting game (also quite naturally called, in an earlier paper that mostly looked at two candidates, the roll-call voting game [Slo93]). This basically is an election in which the voters vote in order, *and the preferences are common knowledge—everyone knows everyone else's preferences, everyone knows that everyone knows everyone else's preferences, and so on out to infinity*. Their analysis of this game is fundamentally game-theoretic; with such complete knowledge in a sequential setting, there is precisely one (subgame perfect Nash) equilibrium, which can be computed from the back end forward. Under their work's setting and assumptions, for bounded numbers of manipulators manipulation is in P, but we will show that in our model even with bounded numbers of manipulators manipulation sometimes (unless P = NP) falls beyond P.

The interesting "dynamic voting" work of Tennenholtz [Ten04] investigates sequential voting, but focuses on axioms and voting rules rather than on coalitions and manipulation. Much heavily Markovian work studies sequential decision-making and/or dynamically varying preferences (see [PP11] and the references therein); our work in contrast is nonprobabilistic and focused on the complexity of coalitional manipulation. Also somewhat related to, but quite different from, our work is the work on possible and necessary winners. The seminal paper on that is due to Konczak and Lang [KL05], and more recent work includes [XC08, BHN09, BBF10, Bet10, BD10, CLM+12, BR12]; the biggest difference is that those are, loosely, one-quantifier settings, but the more dynamic setting of online manipulation involves numbers of quantifiers that can grow with the input size. Another related research line studies multi-issue elections [XC10b, XCL10, XCL11, XLC11]; although there the separate issues may run in sequence, each issue typically is voted on simultaneously and with preferences being common knowledge.

2 Preliminaries

Elections. A *(standard, i.e., simultaneous) election* (C,V) is specified by a set C of candidates and a list V, where we assume that each element in V is a pair (v,p) such that v is a voter name and p is v's vote. How the votes in V are represented depends on the election system used—we assume, as is required by most systems, votes to be total preference orders over C. For example, if $C = \{a,b,c\}$, a vote of the form $c > a > b$ means that this voter (strictly) prefers c to a and a to b.

We introduce election snapshots to capture sequential election scenarios as follows. Let C be a set of candidates and let u be (the name of) a voter. An *election snapshot for C and u* is specified by a triple $V = (V_{<u}, u, V_{u<})$ consisting of all voters in the order they vote, along with, for each voter before u (i.e., those in $V_{<u}$), the vote she cast, and for each voter after u (i.e., those in $V_{u<}$), a bit specifying if she is part of the manipulative coalition (to which u always belongs). That is, $V_{<u} = ((v_1,p_1),(v_2,p_2),\ldots,(v_{i-1},p_{i-1}))$, where the voters named $v_1,v_2,\ldots,v_{i-1}$ (including perhaps manipulators and nonmanipulators) have already cast their votes (preference order p_j being cast by v_j), and $V_{u<} = ((v_{i+1},x_{i+1}),(v_{i+2},x_{i+2}),\ldots,(v_n,x_n))$ lists the names of the voters still to cast their votes, in that order, and where $x_j = 1$ if v_j belongs to the manipulative coalition and $x_j = 0$ otherwise.

Scoring Rules. A *scoring rule* for m candidates is given by a scoring vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$ of nonnegative integers such that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$. For an election (C,V), each candidate $c \in C$ scores α_i points for each vote that ranks c in the ith position. Let $score(c)$ be the total score of $c \in C$. All candidates scoring the most points are winners of (C,V). Some of the most popular voting systems are *k-approval* (especially *plurality*, aka 1-approval) and *k-veto* (especially *veto*, aka 1-veto). Their m-candidate, $m \geq k$, versions are defined by the scoring vectors $(\underbrace{1,\dots,1}_{k},\underbrace{0,\dots,0}_{m-k})$ and $(\underbrace{1,\dots,1}_{m-k},\underbrace{0,\dots,0}_{k})$. When m is not fixed, we omit the phrase "m-candidate."

Manipulation. The *(standard) weighted coalitional manipulation problem* [CSL07], $\mathscr{E}$-Weighted-Coalitional-Manipulation (abbreviated by $\mathscr{E}$-WCM), for any election system $\mathscr{E}$ is defined as follows: Given a candidate set C, a list S of nonmanipulative voters each having a nonnegative integer weight, a list T of the nonnegative integer weights of the manipulative voters (whose preferences over C are unspecified), with $S \cap T = \emptyset$, and a distinguished candidate $c \in C$, can the manipulative votes T be set such that c is a (or the) $\mathscr{E}$ winner of $(C, S \cup T)$?

Asking whether c can be made "a winner" is called the nonunique-winner model and is the model of all notions in this paper unless mentioned otherwise. If one asks whether c can be made a "one and only winner," that is called the unique-winner model. We also use the *unweighted* variant, where each vote has unit weight, and write $\mathscr{E}$-UCM as a shorthand. Note that $\mathscr{E}$-UCM with a *single* manipulator (i.e., $\|T\| = 1$ in the problem instance) is the manipulation problem originally studied in [BTT89, BO91]. Conitzer, Sandholm, and Lang [CSL07] also introduced the *destructive* variants of these manipulation problems, where the goal is not to make c win but to ensure that c is not a winner, and we denote the corresponding problems by $\mathscr{E}$-DWCM and $\mathscr{E}$-DUCM. Finally, we write $\mathscr{E}$-WC$_{\neq\emptyset}$M, $\mathscr{E}$-UC$_{\neq\emptyset}$M, $\mathscr{E}$-DWC$_{\neq\emptyset}$M, and $\mathscr{E}$-DUC$_{\neq\emptyset}$M to indicate that the problem instances are required to have a nonempty coalition of manipulators.

Complexity-Theoretic Background. We assume the reader is familiar with basic complexity-theoretic notions such as the complexity classes P and NP, the class FP of polynomial-time computable functions, polynomial-time many-one reducibility ($\leq_m^p$), and hardness and completeness with respect to $\leq_m^p$ for a complexity class.

Meyer and Stockmeyer [MS72] and Stockmeyer [Sto76] introduced and studied the polynomial hierarchy, $\mathrm{PH} = \bigcup_{k\geq 0} \Sigma_k^p$, whose levels are inductively defined by $\Sigma_0^p = \mathrm{P}$ and $\Sigma_{k+1}^p = \mathrm{NP}^{\Sigma_k^p}$, and their co-classes, $\Pi_k^p = \mathrm{co}\Sigma_k^p$ for $k \geq 0$. They also characterized these levels by polynomially length-bounded alternating existential and universal quantifiers. P^{NP} is the class of problems solvable in deterministic polynomial time with access to an NP oracle, and $\mathrm{P}^{\mathrm{NP}[1]}$ is the restriction of P^{NP} where only one oracle query is allowed. Note that $\mathrm{P} \subseteq \mathrm{NP} \cap \mathrm{coNP} \subseteq \mathrm{NP} \cup \mathrm{coNP} \subseteq \mathrm{P}^{\mathrm{NP}[1]} \subseteq \mathrm{P}^{\mathrm{NP}} \subseteq \Sigma_2^p \cap \Pi_2^p \subseteq \Sigma_2^p \cup \Pi_2^p \subseteq \mathrm{PH} \subseteq \mathrm{PSPACE}$, where PSPACE is the class of problems solvable in polynomial space. The *quantified boolean formula problem*, QBF, is a standard PSPACE-complete problem. Define QBF_k ($\widetilde{\mathrm{QBF}}_k$) to be the restriction of QBF with at most k quantifiers that start with $\exists$ ($\forall$) and then alternate between $\exists$ and $\forall$, and we assume that each $\exists$ and $\forall$ quantifies over a set of boolean variables. For each $k \geq 1$, QBF_k is Σ_k^p-complete and $\widetilde{\mathrm{QBF}}_k$ is Π_k^p-complete.

Proofs omitted due to space limitations can be found in the technical report version [HHR12a].

3 Our Model of Online Manipulation

The core of our model of online manipulation in sequential voting is what we call the *magnifying-glass moment*, namely, the moment at which a manipulator u is the one who is going to vote, is aware of what has happened so far in the election (and which voters are still to come, but in general not knowing what they want, except in the case of voters, if any, who are coalitionally linked to u). In this moment, u seeks to "figure out" what the "best" vote to cast is. We will call the information available in such a moment an *online manipulation setting* (*OMS*, for short) and define it formally as a tuple (C,u,V,σ,d), where C is a set of candidates; u is a distinguished voter; $V = (V_{<u},u,V_{u<})$ is an election snapshot for C and u; σ is the preference order of the manipulative coalition to which u belongs; and $d \in C$ is a distinguished candidate. Given an election system $\mathscr{E}$, define the problem online-$\mathscr{E}$-Unweighted-Coalitional-Manipulation (abbreviated by online-$\mathscr{E}$-UCM), as follows: Given an OMS (C,u,V,σ,d) as described above, does there exist some vote that u can cast (assuming support from the manipulators coming after u) such that no matter what votes are cast by the nonmanipulators coming after u, there exists some $c \in C$ such that $c \geq_\sigma d$ and c is an $\mathscr{E}$ winner of the election? By "support from the manipulators coming after u" we mean that u's coalition partners coming after u, when they get to vote, will use their then-in-hand knowledge of all votes up to then to help u reach her goal: By a joint effort u's coalition can ensure that the $\mathscr{E}$ winner set will always include a candidate liked by the coalition as much as or more than d, even when the nonmanipulators take their strongest action so as to prevent this. Note that this candidate, c in the problem description, may be different based on the nonmanipulators' actions. (Nonsequential manipulation problems usually focus on whether a single candidate can be made to win, but in our setting, this "that person or better" focus is more natural.) For the case of weighted manipulation, each voter also comes with a nonnegative integer weight. We denote this problem by online-$\mathscr{E}$-WCM.

We write online-$\mathscr{E}$-UCM$[k]$ in the unweighted case and online-$\mathscr{E}$-WCM$[k]$ in the weighted case to denote the problem when the number of manipulators from u onward is restricted to be at most k.

Our corresponding destructive problems are denoted by online-$\mathscr{E}$-DUCM, online-$\mathscr{E}$-DWCM, online-$\mathscr{E}$-DUCM$[k]$, and online-$\mathscr{E}$-DWCM$[k]$. In online-$\mathscr{E}$-DUCM we ask whether the given current manipulator u (assuming support from the manipulators after her) can cast a vote such that no matter what votes are cast by the nonmanipulators after u, no $c \in C$ with $d \geq_\sigma c$ is an $\mathscr{E}$ winner of the election, i.e., u's coalition can ensure that the $\mathscr{E}$ winner set never includes d or any even more hated candidate. The other three problems are defined analogously.

Note that online-$\mathscr{E}$-UCM generalizes the original unweighted manipulation problem with a single manipulator as introduced by Bartholdi, Tovey, and Trick [BTT89]. Indeed, their manipulation problem in effect is the special case of online-$\mathscr{E}$-UCM when restricted to instances where there is just one manipulator, she is the last voter to cast a vote, and d is the coalition's most preferred candidate. Similarly, online-$\mathscr{E}$-WCM generalizes the (standard) coalitional weighted manipulation problem (for nonempty coalitions of manipulators). Indeed, that traditional manipulation problem is the special case of online-$\mathscr{E}$-WCM, restricted to instances where only manipulators come after u and d is the coalition's most preferred candidate. If we take an analogous approach except with d restricted now to being the most hated candidate of the coalition, we generalize the corresponding notions for the destructive cases. We summarize these observations as follows.

Proposition 1 *For each election system $\mathscr{E}$, it holds that (1)* $\mathscr{E}$-UC$_{\neq\emptyset}$M $\leq_m^p$ online-$\mathscr{E}$-UCM, *(2)* $\mathscr{E}$-WC$_{\neq\emptyset}$M $\leq_m^p$ online-$\mathscr{E}$-WCM, *(3)* $\mathscr{E}$-DUC$_{\neq\emptyset}$M $\leq_m^p$ online-$\mathscr{E}$-DUCM, *and (4)* $\mathscr{E}$-DWC$_{\neq\emptyset}$M $\leq_m^p$ online-$\mathscr{E}$-DWCM.

Corollary 2 below follows immediately from the above proposition.

Corollary 2 *(1) For each election system $\mathscr{E}$ such that the (unweighted) winner problem is solvable in polynomial time, it holds that* $\mathscr{E}$-UCM $\leq_m^p$ online-$\mathscr{E}$-UCM. *(2) For each election system $\mathscr{E}$ such that the weighted winner problem is solvable in polynomial time, it holds that* $\mathscr{E}$-WCM $\leq_m^p$ online-$\mathscr{E}$-WCM. *(3) For each election system $\mathscr{E}$ such that the winner problem is solvable in polynomial time, it holds that* $\mathscr{E}$-DUCM $\leq_m^p$ online-$\mathscr{E}$-DUCM. *(4) For each election system $\mathscr{E}$ such that the weighted winner problem is solvable in polynomial time, it holds that* $\mathscr{E}$-DWCM $\leq_m^p$ online-$\mathscr{E}$-DWCM.

We said above that, by default, we will use the *nonunique-winner model* and all the above problems are defined in this model. However, we will also have some results in the *unique-winner model*, which will, here, sharply contrast with the corresponding results in the nonunique-winner model. To indicate that a problem, such as online-$\mathscr{E}$-UCM, is in the unique-winner model, we write online-$\mathscr{E}$-UCM$_{\mathrm{UW}}$ and ask whether the current manipulator u (assuming support from the manipulators coming after her) can ensure that there exists some $c \in C$ such that $c \geq_\sigma d$ and c is *the unique* $\mathscr{E}$ *winner* of the election.

4 General Results

Theorem 3 *(1) For each election system $\mathscr{E}$ whose weighted winner problem can be solved in polynomial time,*[1] *the problem* online-$\mathscr{E}$-WCM *is in* PSPACE. *(2) For each election system $\mathscr{E}$ whose winner problem can be solved in polynomial time, the problem* online-$\mathscr{E}$-UCM *is in* PSPACE. *(3) There exists an election system $\mathscr{E}$ with a polynomial-time winner problem such that the problem* online-$\mathscr{E}$-UCM *is* PSPACE-*complete. (4) There exists an election system $\mathscr{E}$ with a polynomial-time weighted winner problem such that the problem* online-$\mathscr{E}$-WCM *is* PSPACE-*complete.*

PROOF. The proof of the first statement (which is analogous to the proof of the first statement in Theorem 4) follows from the easy fact that online-$\mathscr{E}$-WCM can be solved by an alternating Turing machine in polynomial time, and thus, due to the characterization of Chandra, Kozen, and Stockmeyer [CKS81], by a deterministic Turing machine in polynomial space. The proof of the second case is analogous.

We construct an election system $\mathscr{E}$ establishing the third statement. Let (C,u,V,σ,d) be a given input. $\mathscr{E}$ will look at the lexicographically least candidate name in C. Let c represent that name string in some fixed, natural encoding. $\mathscr{E}$ will check if c represents a *tiered* boolean formula, by which we mean one whose variable names are all of the form $x_{i,j}$ (which really means a direct encoding of a string, such as "$x_{4,9}$"); the i,j fields must all be positive integers. If c does not represent such a tiered formula, everyone loses on that input. Otherwise (i.e., if c represents a tiered formula), let *width* be the maximum j occurring as the second subscript in any variable name ($x_{i,j}$) in c, and let *blocks* be the maximum i occurring as the first subscript in any variable name in c. If there are fewer than *blocks* voters in V, everyone loses. Otherwise, if there are fewer than $1+2\cdot \mathit{width}$ candidates in C, everyone loses (this is so that each vote will involve enough candidates that it can be used to set all the variables in one block). Otherwise, if there exists some i, $1 \leq i \leq \mathit{blocks}$, such that for no j does the variable $x_{i,j}$ occur in c, then everyone loses. Otherwise, order the voters from the lexicographically least to the lexicographically greatest voter name. If distinct voters are allowed to have the same name string (e.g., John Smith), we break ties by sorting according to the associated preference orders within each group of tied voters (second-order ties are no problem, as those votes are identical, so any order will have the same effect). Now, the first voter in this order will assign truth values to all variables $x_{1,*}$, the second voter in this order will assign truth values to all variables $x_{2,*}$, and so on up to the *blocks*th voter, who will assign truth values to all variables $x_{\mathit{blocks},*}$.

How do we get those assignments from these votes? Consider a vote whose total order over C is σ' (and recall that $\|C\| \geq 1+2\cdot \mathit{width}$). Remove c from σ', yielding σ''. Let $c_1 <_{\sigma''} c_2 <_{\sigma''} \cdots <_{\sigma''} c_{2\cdot \mathit{width}}$ be the $2\cdot \mathit{width}$ least preferred candidates in σ''. We build a vector in $\{0,1\}^{\mathit{width}}$ as follows: The ℓth bit of the vector is 0 if the string that names $c_{1+2(\ell-1)}$ is lexicographically less than the string that names $c_{2\ell}$, and this bit is 1 otherwise.

Let b_i denote the vector thus built from the ith vote (in the above ordering), $1 \leq i \leq \mathit{blocks}$. Now, for each variable $x_{i,j}$ occurring in c, assign to it the value of the jth bit of b_i, where 0 represents *false*

[1]We mention in passing here, and henceforward we will not explicitly mention it in the analogous cases, that the claim clearly remains true even when "polynomial time" is replaced by the larger class "polynomial space."

and 1 represents *true*. We have now assigned all variables of c, so c evaluates to either *true* or *false*. If c evaluates to *true*, everyone wins, otherwise everyone loses. This completes the specification of the election system $\mathscr{E}$. $\mathscr{E}$ has a polynomial-time winner problem, as any boolean formula, given an assignment to all its variables, can easily be evaluated in polynomial time.

To show PSPACE-hardness, we $\leq_m^p$-reduce the PSPACE-complete problem QBF to the problem online-$\mathscr{E}$-UCM. Let y be an instance of QBF. We transform y into an instance of the form

$$(\exists x_{1,1}, x_{1,2}, \ldots, x_{1,k_1})\,(\forall x_{2,1}, x_{2,2}, \ldots, x_{2,k_2}) \cdots (Q_\ell\, x_{\ell,1}, x_{\ell,2}, \ldots, x_{\ell,k_\ell})$$
$$[\Phi(x_{1,1}, x_{1,2}, \ldots, x_{1,k_1}, x_{2,1}, x_{2,2}, \ldots, x_{2,k_2}, \ldots, x_{\ell,1}, x_{\ell,2}, \ldots, x_{\ell,k_\ell})]$$

in polynomial time, where $Q_\ell = \exists$ if ℓ is odd and $Q_\ell = \forall$ if ℓ is even, the $x_{i,j}$ are boolean variables, Φ is a boolean formula, and for each i, $1 \leq i \leq \ell$, Φ contains at least one variable of the form $x_{i,*}$. This quantified boolean formula is $\leq_m^p$-reduced to an instance (C, u, V, σ, c) of online-$\mathscr{E}$-UCM as follows:

1. C contains a candidate whose name, c, encodes Φ, and in addition C contains $2 \cdot \max(k_1, \ldots, k_\ell)$ other candidates, all with names lexicographically greater than c—for specificity, let us say their names are the $2 \cdot \max(k_1, \ldots, k_\ell)$ strings that immediately follow c in lexicographic order.

2. V contains ℓ voters, $1, 2, \ldots, \ell$, who vote in that order, where $u = 1$ is the distinguished voter and all odd voters belong to u's manipulative coalition and all even voters do not. The voter names will be lexicographically ordered by their number, 1 is least and ℓ is greatest.

3. The manipulators' preference order σ is to like candidates in the opposite of their lexicographic order. In particular, c is the coalition's most preferred candidate.

This is a polynomial-time reduction. It follows immediately from this construction and the definition of $\mathscr{E}$ that y is in QBF if and only if (C, u, V, σ, c) is in online-$\mathscr{E}$-UCM.

To prove the last statement, simply let $\mathscr{E}$ be the election system that ignores the weights of the voters and then works exactly as the previous election system. ❑

The following theorem shows that for bounded numbers of manipulators the complexity crawls up the polynomial hierarchy. The theorem's proof is based on the proof given above, except we need to use the alternating quantifier characterization due to Meyer and Stockmeyer [MS72] and Stockmeyer [Sto76] for the upper bound and to reduce from the Σ_{2k}^p-complete problem QBF_{2k} rather than from QBF for the lower bound.

Theorem 4 *Fix any $k \geq 1$. (1) For each election system $\mathscr{E}$ whose weighted winner problem can be solved in polynomial time, the problem* online-$\mathscr{E}$-WCM$[k]$ *is in Σ_{2k}^p. (2) For each election system $\mathscr{E}$ whose winner problem can be solved in polynomial time, the problem* online-$\mathscr{E}$-UCM$[k]$ *is in Σ_{2k}^p. (3) There exists an election system $\mathscr{E}$ with a polynomial-time winner problem such that the problem* online-$\mathscr{E}$-UCM$[k]$ *is Σ_{2k}^p-complete. (4) There exists an election system $\mathscr{E}$ with a polynomial-time weighted winner problem such that the problem* online-$\mathscr{E}$-WCM$[k]$ *is Σ_{2k}^p-complete.*

Note that the (constructive) online manipulation problems considered in Theorems 3 and 4 are about ensuring that the winner set always contains some candidate in the σ segment stretching from d up to the top-choice. Now consider "pinpoint" variants of these problems, where we ask whether the distinguished candidate d herself can be guaranteed to be a winner (for nonsequential manipulation, that version indeed is the one commonly studied). Denote the *pinpoint* variant of, e.g., online-$\mathscr{E}$-UCM$[k]$ by pinpoint-online-$\mathscr{E}$-UCM$[k]$. Since our hardness proofs in Theorems 3 and 4 make all or no one a winner (and as the upper bounds in these theorems also can be seen to hold for the pinpoint variants), they establish the corresponding completeness results also for the pinpoint cases. We thus have completeness results for PSPACE and Σ_{2k}^p for each $k \geq 1$. What about the classes Σ_{2k-1}^p and Π_k^p, for each $k \geq 1$? We can get completeness results for all these classes by defining appropriate variants of online manipulation problems. Let OMP be any of the online manipulation problems considered earlier, including the pinpoint variants mentioned above. Define freeform-OMP to be just as OMP, except we no longer require the distinguished voter u to be part of the manipulative coalition—u can be in or can be out, and the input must specify, for u and all voters after u, which ones are the members of the coalition. The question of freeform-OMP is whether it is true that for all actions of the nonmanipulators at or after u (for specificity as to this problem: if u is a nonmanipulator, it will in the input come with a preference order) there will be actions (each taken with full information on cast-before-them votes) of the manipulative coalition members such that their goal of making some candidate c with $c \geq_\sigma d$ (or exactly d, in the pinpoint versions) a winner is achieved. Then, whenever Theorem 4 establishes a Σ_{2k}^p or Σ_{2k}^p-completeness result for OMP, we obtain a Π_{2k+1}^p or Π_{2k+1}^p-completeness result for freeform-OMP and for $k = 0$ manipulators we obtain $\Pi_1^p = \text{coNP}$ or coNP-completeness results. Similarly, the PSPACE and PSPACE-completeness results for OMP we established in Theorem 3 also can be shown true for freeform-OMP.

On the other hand, if we define a variant of OMP by requiring the final voter to always be a manipulator, the PSPACE and PSPACE-completeness results for OMP from Theorem 3 remain true for this variant; the Σ_{2k}^p and Σ_{2k}^p-completeness results for OMP from Theorem 4 change to Σ_{2k-1}^p and Σ_{2k-1}^p-completeness results for this variant; and the above Π_{2k+1}^p and Π_{2k+1}^p-completeness results for freeform-OMP change to Π_{2k}^p and Π_{2k}^p-completeness results for this variant, $k \geq 1$.

Finally, as an open direction (and related conjecture), we define for each of the previously considered variants of online manipulation problems a *full profile* version. For example, for a given election system $\mathscr{E}$, fullprofile-online-$\mathscr{E}$-UCM$[k]$ is the function problem that, given an OMS *without* any distinguished candidate, (C,u,V,σ), returns a length $\|C\|$ bit-vector that for each candidate $d \in C$ says if the answer to "$(C,u,V,\sigma,d) \in$ online-$\mathscr{E}$-UCM$[k]$?" is "yes" (1) or "no" (0). The function problem fullprofile-pinpoint-online-$\mathscr{E}$-UCM$[k]$ is defined analogously, except regarding pinpoint-online-$\mathscr{E}$-UCM$[k]$.

It is not hard to prove, as a corollary to Theorem 4, that:

Theorem 5 *For each election system $\mathscr{E}$ whose winner problem can be solved in polynomial time, (1)* fullprofile-online-$\mathscr{E}$-UCM$[k]$ *is in* $\text{FP}^{\Sigma_{2k}^p[\mathscr{O}(\log n)]}$*, the class of functions computable in polynomial time given Turing access to a Σ_{2k}^p oracle with $\mathscr{O}(\log n)$ queries allowed on inputs of size n, and (2)* fullprofile-pinpoint-online-$\mathscr{E}$-UCM$[k]$ *is in* $\text{FP}_{\text{tt}}^{\Sigma_{2k}^p}$*, the class of functions computable in polynomial time given truth-table access to a Σ_{2k}^p oracle.*

We conjecture that both problems are complete for the corresponding class under metric reductions [Kre88], for suitably defined election systems with polynomial-time winner problems.

If the full profile version of an online manipulation problem can be computed efficiently, we clearly can also easily solve each of the decision problems involved by looking at the corresponding bit of the length $\|C\|$ bit-vector. Conversely, if there is an efficient algorithm for an online manipulation decision problem, we can easily solve its full profile version by running this algorithm for each candidate in turn. Thus, we will state our later results only for online manipulation decision problem.

Proposition 6 *Let* OMP *be any of the online manipulation decision problems defined above. Then* fullprofile-OMP *is in* FP *if and only if* OMP *is in* P.

5 Results for Specific Natural Voting Systems

The results of the previous section show that, simply put, even for election systems with polynomial-time winner problems, online manipulation can be tremendously difficult. But what about *natural* election systems? We will now take a closer look at important natural systems. We will show that online manipulation can be easy for them, depending on which particular problem is considered, and we will also see that the constructive and destructive cases can differ sharply from each other and that it really matters whether we are in the nonunique-winner model or the unique-winner model.

Theorem 7 *(1)* online-plurality-WCM *(and thus also* online-plurality-UCM*) is in* P. *(2)* online-plurality-DWCM *(and thus also* online-plurality-DUCM*) is in* P.

Theorem 7 refers to problems in the nonunique-winner model. By contrast, we now show that online manipulation for weighted plurality voting in the *unique-winner* model is coNP-hard in the *constructive* case and is NP-hard in the *destructive* case. This is perhaps the most dramatic, broad contrast yet between the nonunique-winner model and the unique-winner model, and is the first such contrast involving plurality. The key other NP-hardness versus P result for the nonunique-winner model versus the unique-winner model is due to Faliszewski, Hemaspaandra, and Schnoor [FHS08], but holds only for (standard) weighted manipulation for Copeland$^{\alpha}$ elections ($0 < \alpha < 1$) with exactly three candidates; for fewer than three both cases there are in P and for more than three both are NP-complete. In contrast, the P results of Theorem 7 hold for all numbers of candidates, and the NP-hardness and coNP-hardness results of Theorem 8 hold whenever there are at least two candidates.

Theorem 8 *(1) The problem* online-plurality-DWCM$_{\text{UW}}$ *is* NP*-hard, even when restricted to only two candidates (and this also holds when restricted to three, four, ... candidates). (2) The problem* online-plurality-WCM$_{\text{UW}}$ *is* coNP*-hard, even when restricted to only two candidates (and this also holds when restricted to three, four, ... candidates).*

PROOF. For the first statement, we prove NP-hardness of online-plurality-DWCM$_{\text{UW}}$ by a reduction from the NP-complete problem Partition: Given a nonempty sequence $(w_1, w_2, \ldots, w_z)$ of positive integers such that $\sum_{i=1}^{z} w_i = 2W$ for some positive integer W, does there exist a set $I \subseteq \{1, 2, \ldots, z\}$ such that $\sum_{i \in I} w_i = W$? Let $m \geq 2$. Given an instance $(w_1, w_2, \ldots, w_z)$ of Partition, construct an instance $(\{c_1, \ldots, c_m\}, u_1, V, c_1 > c_2 > \cdots > c_m, c_1)$ of online-plurality-DWCM$_{\text{UW}}$ such that V contains $m + z - 2$ voters $v_1, \ldots, v_{m-2}, u_1, \ldots, u_z$ who vote in that order. For $1 \leq i \leq m-2$, v_i votes for c_i and has weight $(m-1)W - i$, and for $1 \leq i \leq z$, u_i is a manipulator of weight $(m-1)w_i$. If $(w_1, w_2, \ldots, w_z)$ is a yes-instance of Partition, the manipulators can give $(m-1)W$ points to both c_{m-1} and c_m, and zero points to the other candidates. So c_{m-1} and c_m are tied for the most points and there is no unique winner. On the other hand, the only way to avoid having a unique winner in our online-plurality-DWCM$_{\text{UW}}$ instance is if there is a tie for the most points. The only candidates that can tie are c_{m-1} and c_m, since all other pairs of candidates have different scores modulo $m-1$. It is easy to see that c_{m-1} and c_m tie for the most points only if they both get exactly $(m-1)W$ points. It follows that $(w_1, w_2, \ldots, w_z)$ is a yes-instance of Partition.

For the second part, we adapt the above construction to yield a reduction from Partition to the complement of online-plurality-WCM$_{\text{UW}}$. Given an instance $(w_1, w_2, \ldots, w_z)$ of Partition, construct an instance $(\{c_1, \ldots, c_m\}, \widehat{u}, V, c_1 > c_2 > \cdots > c_m, c_m)$ of online-plurality-WCM$_{\text{UW}}$ such that V contains $m + z - 1$ voters $v_1, \ldots, v_{m-2}, \widehat{u}, u_1, \ldots, u_z$ who vote in that order. For $1 \leq i \leq m-2$, v_i has the same vote and the same weight as above, $\widehat{u}$ is a manipulator of weight 0, and for $1 \leq i \leq z$, u_i has the same weight as above, but in contrast to the case above, u_i is now a nonmanipulator. By the same argument as above, it follows that $(w_1, w_2, \ldots, w_z)$ is a yes-instance of Partition if and only if the nonmanipulators can ensure that there is no unique winner, which in turn is true if and only if the manipulator can not ensure that there is a unique winner. ❑

Theorem 9 *For each scoring rule* $\alpha = (\alpha_1, \ldots, \alpha_m)$, online-$\alpha$-WCM *is in* P *if* $\alpha_2 = \alpha_m$ *and is* NP*-hard otherwise.*

Theorem 10 *For each k,* online-k-approval-UCM *and* online-k-veto-UCM *are in* P.

PROOF. Consider 1-veto. Given an online-1-veto-UCM instance (C,u,V,σ,d), the best strategy for the manipulators from u onward (let n_1 denote how many of these there are) is to minimize $\max_{c<_\sigma d} score(c)$. Let n_0 denote how many nonmanipulators come after u. We claim that (C,u,V,σ,d) is a yes-instance if and only if d is ranked last in σ or there exists a threshold t such that (1) $\sum_{c<_\sigma d}(maxscore(c) \ominus t) \leq n_1$ (so those manipulators can ensure that all candidates ranked $<_\sigma d$ score at most t points), where "$\ominus$" denotes proper subtraction ($x \ominus y = \max(x-y,0)$) and $maxscore(c)$ is c's score when none of the voters from u onward veto c, and (2) $\sum_{c\geq_\sigma d}(maxscore(c) \ominus (t-1)) > n_0$ (so those nonmanipulators cannot prevent that some candidate ranked $\geq_\sigma d$ scores at least t points).

For 1-veto under the above approach, in each situation where the remaining manipulators can force success against all actions of the remaining nonmanipulators, u (right then as she moves) can set her *and all future manipulators' actions* so as to force success regardless of the actions of the remaining nonmanipulators. For k-approval and k-veto, $k \geq 2$, that approach provably cannot work (as will be explained right after this proof); rather, we sometimes need later manipulators' actions to be shaped by intervening nonmanipulators' actions. Still, the following P-time algorithm, which works for all k, tells whether success can be forced. As a thought experiment, for each voter v from u onwards in sequence do this: Order the candidates in $\{c \mid c \geq_\sigma d\}$ from most to least current approvals, breaking ties arbitrarily, and postpend the remaining candidates ordered from least to most current approvals. Let ℓ be k for k-approval and $\|C\|-k$ for k-veto. Cast the voter's ℓ approvals for the first ℓ candidates in this order if v is a manipulator, and otherwise for the last ℓ candidates in this order. Success can be forced against perfect play if and only if this P-time process leads to success. ❑

In the above proof we said that the approach for 1-veto (in which the current manipulator can set her and all future manipulators' actions so as to force success independent of the actions of intervening future nonmanipulators) provably cannot work for k-approval and k-veto, $k \geq 2$. Why not? Consider an OMS (C,u,V,σ,d) with candidate set $C = \{c_1,c_2,\dots,c_{2k}\}$, σ being given by $c_1 >_\sigma c_2 >_\sigma \cdots >_\sigma c_{2k}$, and $d = c_1$. So, u's coalition wants to enforce that c_1 is a winner. Suppose that v_1 has already cast her vote, now it's $v_2 = u$'s turn, and the order of the future voters is $v_3,v_4,\dots,v_{2j}$, where all v_{2i}, $2 \leq i \leq j$, belong to u's coalition, and all v_{2i-1} do not. Suppose that v_1 was approving of the k candidates in $C_1 \subseteq \{c_2,c_3,\dots,c_{2k}\}$, $\|C_1\| = k$. Then u must approve of the k candidates in $\overline{C_1}$, to ensure that c_1 draws level with the candidates in C_1 and none of these candidates can gain another point. Next, suppose that nonmanipulator v_3 approves of the k candidates in $C_3 \subseteq \{c_2,c_3,\dots,c_{2k}\}$, $\|C_3\| = k$. Then v_4, the next manipulator, must approve of all candidates in $\overline{C_3}$, to ensure that c_1 draws level with the candidates in C_3 and none of these candidates can gain another point. This process is repeated until the last nonmanipulator, v_{2j-1}, approves of the candidates in $C_{2j-1} \subseteq \{c_2,c_3,\dots,c_{2k}\}$, $\|C_{2j-1}\| = k$, and v_{2j}, the final manipulator, is forced to counter this by approving of all candidates in $\overline{C_{2j-1}}$, to ensure that c_1 is a winner. This shows that there can be arbitrarily long chains such that the action of each manipulator after u depends on the action of the preceding intervening nonmanipulator.

We now turn to online weighted manipulation for veto when restricted to three candidates. We denote this restriction of online-veto-WCM by online-veto$_{|3}$-WCM.

Theorem 11 online-veto$_{|3}$-WCM *is* $\mathrm{P}^{\mathrm{NP}[1]}$*-complete.*

Dropping the restriction to three candidates, we obtain the following result, which places this problem far below the general PSPACE bound from earlier in this paper. Immediately from Theorems 10 and 12, we have that the full profile variants of online-k-veto-UCM and online-k-approval-UCM are in FP and that fullprofile-online-veto-WCM is in $\mathrm{FP}^{\mathrm{NP}}$.

Theorem 12 online-veto-WCM *is in* P^{NP}.

6 Uncertainty About the Order of Future Voters

So far, we have been dealing with cases where the order of future voters was fixed and known. But what happens if the order of future voters itself is unknown? Even here, we can make claims. To model this most naturally, our "magnifying-glass moment" will focus not on one manipulator u, but will focus at a moment in time when some voters are still to come (as before, we know who they are and which are manipulators; as before, we have a preference order σ, and know what votes have been cast so far, and have a distinguished candidate d). And the question our problem is asking is: Is it the case that our manipulative coalition can ensure that the winner set will always include d or someone liked more than d with respect to σ (i.e., the winner set will have nonempty intersection with $\{c \in C \mid c \geq_\sigma d\}$), *regardless of what order the remaining voters vote in*. We will call this problem the *schedule-robust online manipulation problem*, and will denote it by SR-online-$\mathscr{E}$-UCM. (We will add a "[1,1]" suffix for the restriction of this problem to instances when at most one manipulator and at most one nonmanipulator have not yet voted.) One might think that this problem captures both a Σ_2^p and a Π_2^p issue, and so would be hard for both classes. However, the requirement of schedule robustness tames the problem (basically what underpins that is simply that exists-forall-predicate implies forall-exists-predicate), bringing it into Σ_2^p. Further, we can prove, by explicit construction of such a system, that for some simple election systems this problem is complete for Σ_2^p.

Theorem 13 *(1) For each election system $\mathscr{E}$ whose winner problem is in* P*,* SR-online-$\mathscr{E}$-UCM *is in Σ_2^p. (2) There exists an election system $\mathscr{E}$, whose winner problem is in* P*, such that* SR-online-$\mathscr{E}$-UCM *(indeed, even* SR-online-$\mathscr{E}$-UCM$[1,1]$*) is Σ_2^p-complete.*

7 Conclusions and Open Questions

We introduced a novel framework for online manipulation in sequential voting, and showed that manipulation there can be tremendously complex even for systems with simple winner problems. We also showed that among the most important election systems, some have efficient online manipulation algorithms but others (unless $\mathrm{P} = \mathrm{NP}$) do not. It will be important to, complementing our work, conduct typical-case complexity studies. Also, we have extended the scope of our investigation by studying online control [HHR12c, HHR12b] and will do so by studying online bribery in appropriate models.

Acknowledgments: We thank the reviewers for helpful comments. This work was supported in part by NSF grants CCF-0915792, CCF-1101452, and CCF-1101479, ARC grant DP110101792, DFG grant RO-1202/15-1, an SFF grant of HHU Düsseldorf, Friedrich Wilhelm Bessel Research Awards to Edith Hemaspaandra and Lane A. Hemaspaandra from the Alexander von Humboldt Foundation, and a DAAD grant for a PPP project in the PROCOPE program.

References

[BBF10] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 697–702. AAAI Press, July 2010.

[BD10] N. Betzler and B. Dorn. Towards a dichotomy of finding possible winners in elections based on scoring rules. *Journal of Computer and System Sciences*, 76(8):812–836, 2010.

[BE98] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[Bet10] N. Betzler. On problem kernels for possible winner determination under the k-approval protocol. In *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science*, pages 114–125. Springer-Verlag *Lecture Notes in Computer Science #6281*, August 2010.

[BHN09] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 53–58. AAAI Press, July 2009.

[BO91] J. Bartholdi, III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[BR12] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. *Information Processing Letters*, 112(5):186–190, 2012.

[BTT89] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[CKS81] A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *Journal of the ACM*, 26(1), 1981.

[CLM+12] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. *Mathematical Social Sciences*, 64(1):74–88, 2012.

[Con10] V. Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010.

[CSL07] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[DE10] Y. Desmedt and E. Elkind. Equilibria of plurality voting with abstentions. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 347–356. ACM Press, June 2010.

[DKNS01] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference*, pages 613–622. ACM Press, March 2001.

[FHH10] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[FHHR09] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, pages 375–406. Springer, 2009.

[FHS08] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 983–990. International Foundation for Autonomous Agents and Multiagent Systems, May 2008.

[FP10] P. Faliszewski and A. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[Gib73] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[GMHS99] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of recommender systems. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 434–435. ACM Press, 1999.

[HHR12a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of online manipulation of sequential elections. Technical Report arXiv:1202.6655 [cs.GT], Computing Research Repository, arXiv.org/corr/, February 2012. Revised May, 2012.

[HHR12b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Controlling candidate-sequential elections. In *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, August 2012. Short paper. To appear.

[HHR12c] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Online voter control in sequential elections. In *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, August 2012. To appear.

[KL05] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[Kre88] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988.

[MS72] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129. IEEE Press, October 1972.

[PP11] D. Parkes and A. Procaccia. Dynamic social choice: Foundations and algorithms. Working paper, May 2011.

[Sat75] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[Slo93] B. Sloth. The theory of voting and equilibria in noncooperative games. *Games and Economic Behavior*, 5(1):152–169, 1993.

[Sto76] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[Ten04] M. Tennenholtz. Transitive voting. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 230–231. ACM Press, July 2004.

[XC08] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 196–201. AAAI Press, July 2008.

[XC10a] L. Xia and V. Conitzer. Stackelberg voting games: Computational aspects and paradoxes. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 697–702. AAAI Press, July 2010.

[XC10b] L. Xia and V. Conitzer. Strategy-proof voting rules over multi-issue domains with restricted preferences. In *Proceedings of the 6th International Workshop On Internet And Network Economics*, pages 402–414. Springer-Verlag *Lecture Notes in Computer Science #6484*, December 2010.

[XCL10] L. Xia, V. Conitzer, and J. Lang. Aggregating preferences in multi-issue domains by using maximum likelihood estimators. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 399–406. International Foundation for Autonomous Agents and Multiagent Systems, May 2010.

[XCL11] L. Xia, V. Conitzer, and J. Lang. Strategic sequential voting in multi-issue domains and multiple-election paradoxes. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 179–188. ACM Press, 2011.

[XLC11] L. Xia, J. Lang, and V. Conitzer. Hypercubewise preference aggregation in multi-issue domains. In *Proceedings of the 22st International Joint Conference on Artificial Intelligence*, pages 158–163. AAAI Press, July 2011.

Edith Hemaspaandra
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623, USA
Email: `eh@cs.rit.edu`

Lane A. Hemaspaandra
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA
Email: `lane@cs.rochester.edu`

Jörg Rothe
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: `rothe@cs.uni-duesseldorf.de`

Implementation by Agenda Voting

Sean Horan

Abstract

Agenda voting occurs in a wide variety of contexts. This paper characterizes the class of social choice functions that can be implemented by sophisticated voting on an agenda under the assumption of complete information. The main result establishes that a simple *pairwise condition* is necessary and sufficient for implementation by agenda voting.
Keywords: Sophisticated voting, implementation, voting agendas.

1 Introduction

Voting by agenda occurs in a wide variety of political and social choice contexts. The economic analysis of agendas has a rich tradition in the literature dating back to the early work of Black [1958] and Farquharson [1969]. This paper contributes to that literature by characterizing the social choice functions that can be implemented by *sophisticated* voting in an environment with complete information. The main result establishes that a simple *pairwise condition* defined on pairs of states is necessary and sufficient for implementation. The paper builds on earlier work by Srivastava and Trick [1996], who conjectured that a weaker condition defined on pairs of *adjacent* states (i.e. states that differ on the ranking of two outcomes) was necessary and sufficient.[1]

Formally, a voting agenda describes a binary tree where, at any decision node, the agents vote between two collections of competing proposals. Ultimately, the winning proposal is the outcome that survives the sequence of binary votes given by the agenda. If the agents are forward-looking, their behavior is *sophisticated* and the winning proposal can be determined by backward induction. Provided that the provisional winners are determined by simple majority, the winning proposal must be drawn from the *Condorcet set* — the subset of outcomes that indirectly dominates every outcome (see Miller [1977], McKelvey-Niemi [1978], Moulin [1986]). Because the *pairwise condition* is relatively weak, agenda voting is capable of implementing a wide variety of selections from the Condorcet set.

A variety of approaches can be used to implement outcomes from the Condorcet set. Most closely related to implementation by agenda voting are the extensive-form mechanisms based on *backward induction* (Gol'berg-Gourvitch [1986] and Herrero-Srivastava [1992]) and *subgame perfection* (Abreu-Sen [1990], Moore-Repullo [1988], and Vartiainen [2007a]). Also related to implementation by agenda voting are the normal-form solution concepts based on *dominance solvable voting* (Moulin [1979]) and *undominated Nash equilibrium* (Palfrey-Srivastava [1989]).

While each of these four mechanisms is capable of implementing a wider variety of outcomes than agenda voting (see e.g. Dutta-Sen [1993]), there are some compelling advantages to the approach taken here. Perhaps most importantly, agenda voting is a straightforward way to decentralize choice. In contrast with many other approaches to implementation, agenda voting *expressly* rules out artificial features like randomization (see e.g. Vartiainen [2007b]), nuisance strategies (e.g. integer games and *bad* outcomes), and unnaturally complex strategy sets.[2] Arguably, the simplicity of agenda voting is a large part of the reason that this mechanism is so widely used in real-world settings.

No less attractive is the fact that the necessary and sufficient condition for implementation

[1]Their conjecture replaces an earlier conjecture due to Herrero and Srivastava [1992].

[2]However, the agenda required to implement the desired outcomes may be large (see Trick [2006, 2009]).

by agenda voting is simple. By comparison, both of the extensive-form mechanisms discussed above impose necessary conditions that can be quite difficult to verify in practice. In view of this shortcoming, Moore [1992] has stressed the importance of finding a "full and workable characterization of social choice functions that can be implemented in trees." The main result goes some way towards achieving this goal. As discussed, it provides a workable characterization for a fairly broad class of social choice functions that can be implemented in trees. Moreover, it also provides a simple sufficient condition for implementation in trees more generally. This follows from the fact that agenda voting is a special case of implementation via backward induction.

Before moving on, it is worth noting that the approach taken in this paper is somewhat unconventional from a technical standpoint. In the implementation literature, sufficiency of the characterization is generally established by constructing a mechanism that implements any social choice function with the prescribed features. Unusually, the sufficiency proof given here is obtained by algebraic methods that do not rely on the explicit construction of a mechanism. The basic idea of the proof is that extensive-form games can be "added" together at the root to form a new game. The strength of this approach is that the equilibrium of the new game is easily determined from the equilibria of the original games. Since the intuition is straightforward, it is perhaps surprising that very little work in implementation theory leverages the algebraic structure of extensive-form games. The only notable exception is the characterization of implementation via backward induction given by Gol'berg-Gourvitch [1986].

2 Implementation by Agenda Voting

Before stating the main result, this section provides some preliminary definitions and gives some examples of voting agendas that are widely discussed in the literature of social choice and political economy. A discussion of the result is given in Section 3.

2.1 Definitions

Let X denote some finite set of outcomes. The population of agents is given by $A = \{1, ..., a\}$ where $a = |A|$ is odd. Let $\mathcal{L}$ denote the collection of linear orders on X. An element $P{=}(\succ_1, ..., \succ_a)$ of $\mathcal{L}^a$ represents a profile of individual preference orders on X. For any profile $P \in \mathcal{L}^a$, the *majority relation* R is defined by xRy iff $|\{i \in A : x \succ_i y\}| > |\{i \in A : y \succ_i x\}|$. Since $|A|$ is odd, any majority relation R is a complete, asymmetric relation (or *pairwise ranking*) on X. Let $\mathcal{R}$ denote the collection of majority relations on X.

A *social choice function* (SCF) is a mapping $F : \mathcal{L}^a \to X$ that selects an outcome for every profile $P \in \mathcal{L}^a$. A *Condorcet social choice function* is an SCF that selects the *same outcome* when the majority relations on P and P' coincide (i.e. $R = R'$). In other words, it can be described as a mapping $f : \mathcal{R} \to X$ that selects an outcome for every majority relation $R \in \mathcal{R}$. In what follows, I frequently abbreviate by referring to a majority relation R as a *state*.

Generically, a voting agenda can be described as a labelled binary tree. A *binary tree* B is a pair $(V, <)$ consisting of a finite set V of *vertices* and a strict (but incomplete) transitive order $<$ on V. The order $<$ has a particular structure so that: every vertex has either zero or two *successors* and all vertices except one have a unique *predecessor*. The $<$-maximal vertices in V, denoted by V_0, are the *leaves* of the tree and the unique $<$-minimal vertex v^* is the *root*. In order to label the leaves V_0 of a binary tree B with the alternatives in X (where $|X| \leq |V_0|$), let $\iota : V_0 \to X$ define a surjection, or *seeding*, from the leaves to the elements of X. Together, the binary tree B and the seeding ι define a *voting agenda* $T = (B, \iota)$ over the alternatives in X.

For any voting agenda T and majority relation R on X, the *overall winner* $c_T(R) = v^*(T, R)$ is determined by backward induction. The winner $v(T, R)$ at any leaf $v \in V_0$ is the alternative

$\iota(v)$ that labels v and the winner at any non-leaf $v \notin V_0$ is given by majority voting between the winners at the left successor v_l of v and right successor v_r. Formally:

$$v(T, R) \equiv \begin{cases} v_l(T; R) & \text{if } (v_l(T; R), v_r(T; R)) \in R \\ v_r(T; R) & \text{otherwise} \end{cases}$$

Definition 1 (Implementation by Agenda Voting) *A Condorcet SCF f is* ***implementable*** *if there exists an agenda T on X such that $c_T(R) = f(R)$ for every state $R \in \mathcal{R}$.*

Before moving on, I pause to make two comments about this definition. First, observe that it requires implementation for *all* possible states (i.e. every majority relation on X). In the literature, this is known as the *universal domain* assumption. Second, it requires that the agenda implementing f contain *every* alternative in X. Given the assumption of universal domain, this is without loss of generality. The reason is that agenda voting must select the *Condorcet winner* – i.e. the item $x \in X$ s.t. xRy for all $y \in X \setminus \{x\}$ – whenever it exists. Because every alternative in X is the Condorcet winner for some state(s), every alternative is chosen in some state — and, hence, must be part of the agenda.

When there is no Condorcet winner, agenda voting must select from the Condorcet set:

Definition 2 (Condorcet Set) *The* ***Condorcet set*** $\mathbf{C}(Y, R)$ *of the pairwise ranking R on Y is the smallest subset of Y where yRy' for all $y \in \mathbf{C}(Y, R)$ and $y' \in A \backslash \mathbf{C}(Y, R)$. When R is understood, I abbreviate to $\mathbf{C}(Y)$.*

In other words, $\mathbf{C}(Y)$ is a (possibly degenerate) cycle in Y whose members pairwise-dominate every outcome in $Y \backslash \mathbf{C}(Y)$ (see e.g. Moulin [1986] and Laslier [1997]). Intuitively, the Condorcet set generalizes the usual notion of maximization to address the situation where no single outcome R-dominates every other outcome in Y.

2.2 Main Result

The main result establishes that f is implementable if it is implementable for all pairs of majority relations. Formally, outcomes x and x' are *pairwise implementable* in states R and R' if there exists a voting agenda T such that $c_T(R) = x$ and $c_T(R') = x'$. To state the main result:

Main Result *A Condorcet SCF f is implementable iff is implementable for every pair of states.*

Based on the work of Srivastava and Trick [1996], it can be shown that any outcomes x and x' in the Condorcet sets of R and R' are pairwise implementable if the two states are sufficiently distinct on a *global* level. If the states are globally similar however, one can only implement outcomes from the same *locale.* Some definitions are required to formalize these notions.

Given a pairwise-ranking R on Y, a subset $Y' \subseteq Y$ is a *component* of R if every element in Y' bears the same relation to elements in $Y \backslash Y'$. Given an item $y \in Y \backslash Y'$ and any items y', $y'' \in Y'$, then $y''Ry$ if and only if $y'Ry$.[3] A *decomposition* of a pairwise-ranking R on $Y \subseteq X$ is a partition of Y into components. The largest decomposition is the *degenerate partition* $\{Y\}$.

If R is *cyclic* on Y (so that $\mathbf{C}(Y) = Y$), the maximal non-degenerate decomposition $\mathbf{D}(Y, R)$ is unique (see Theorem 1.3.11 of Laslier [1997]). Moreover, the *quotient ranking* $R/\mathbf{D}(Y, R)$ induces a pairwise-ranking on the components of $\mathbf{D}(Y, R)$. Formally, the global structure of a state is determined by the maximal decomposition of the Condorcet set.

[3]To get a better intuition for this definition, note that the Condorcet set is a component of R. In particular, $\mathbf{C}(R, Y)$ is the *smallest component* of R such that $Y \backslash \mathbf{C}(R, Y)$ is also a component of R where cRy for some $c \in \mathbf{C}(R, Y)$ and $y \in Y \backslash \mathbf{C}(R, Y)$.

Definition 3 (Global Structure) *For a pairwise-ranking R on $Y \subseteq X$, the* ***global structure*** $\langle \mathbf{G}(Y), R_G \rangle$ *is a pair consisting of the maximal decomposition* $\mathbf{G}(Y) = \mathbf{D}(\mathbf{C}(Y), R)$ *of R on the Condorcet set and the quotient ranking* $R_G = R/\mathbf{D}(\mathbf{C}(Y), R)$*. Moreover, any component* $g \in \mathbf{G}(Y)$ *defines a* ***locale****.*

States R and R' are *globally distinct* if $\langle \mathbf{G}(X), R_G \rangle \neq \langle \mathbf{G}'(X), R'_G \rangle$ and *globally similar* if $\langle \mathbf{G}(X), R_G \rangle = \langle \mathbf{G}'(X), R'_G \rangle$. In other words, two rankings are similar if they have the same Condorcet set $\mathbf{C}$ and the global structure of the rankings on $\mathbf{C}$ is similar. Conversely, two rankings are distinct when their Condorcet sets differ *or* the global structure of the rankings on $\mathbf{C}$ is distinct.

The condition for implementation on pairs of states can be stated in terms of the global structure. In particular, x and x' are said to be *pairwise implementable* on $Y \subseteq X$ (in states R and R') if there exists a voting agenda T on Y such that $c_T(R) = x$ and $c_T(R') = x'$. Given the main result, the following proposition fully characterizes implementation by agenda voting:

Proposition 1 (Pairwise Condition) ***(I)*** *For globally distinct states R and R', the outcomes x and x' are pairwise implementable iff $x \in \mathbf{C}(X, R)$ and $x' \in \mathbf{C}(X, R')$.* ***(II)*** *For globally similar states R and R', the outcomes x and x' are pairwise implementable iff they are in the same locale $g \in \mathbf{G}(X)$ and are pairwise implementable for some subset $g^* \subseteq g$.*

It is worth clarifying that the main result does not depend on the fact that voting is by majority. More generally, the pairwise-ranking on any profile $P \in \mathcal{P}$ may be derived from any *strong proper simple (SPS) game* (A, W). For a simple game (A, W), the set $W \subseteq 2^a$ defines a *monotonic* collection of winning coalitions such that $w \in W$ and $w \subseteq w'$ imply $w' \in W$. The simple game (A, W) is said to be strong and proper if a coalition w wins whenever its complement $A \backslash w$ loses (so that $w \in W$ iff $A \backslash w \notin W$).

Formally, any SPS game induces a pairwise ranking P_W such that $xP_W y$ iff $\{a \in A : x \succ_a y\} \in W$. When $\mathcal{P}_W \subseteq \mathcal{R}$ is the collection of pairwise rankings induced by the SPS game (A, W), then $f : \mathcal{P}_W \to X$ defines a (partial) Condorcet social choice function. Generalizing the notion of implementation defined above, a social choice function $F : \mathcal{L}^a \to X$ is said to be *implementable* if there exists an SPS game (A, W) and a voting agenda T such that $c_T(P_W) = F(P)$ for any profile $P \in \mathcal{P}$. Thus, F can be implemented by the SPS game (A, W) if and only if f_W is implementable and $F(P) = f_W(P_W)$ for any profile $P \in \mathcal{P}$. As such, the following generalization of the main result is immediate:

Theorem 1 *An SCF F is implementable if there exists an SPS game (A, W) and a (partial) Condorcet social choice function f s.t. (i) f is implementable for every pair of states and, (ii) $F(P) = f(P_W)$ for any $P \in \mathcal{P}$.*

One benefit of using an SPS game different from majority voting is that distinct outcomes may be implemented on profiles whose majority relations coincide. However, it should be kept in mind that departures from majority voting come at the cost of anonymity.

3 Discussion

In earlier work, Srivastava and Trick [1996] showed a necessary and sufficient condition for pairwise implementation on *adjacent states* R and R' that differ only on the pairwise-ranking of two outcomes y and y'. To state their condition, let $B_{R,R'}$ define the smallest component of R such that $\{y, y'\} \subseteq B_{R,R'}$. Srivastava and Trick established that *distinct* outcomes are pairwise implementable on R and R' iff they are in the Condorcet sets of X *and* $B_{R,R'}$ for each state.

Proposition 2 (Adjacent Pairwise Implementation) *x and x' are pairwise implementable on adjacent states R and R' iff: (i) $x \in \mathbf{C}(R, X)$ and $x' \in \mathbf{C}(R', X)$; and, (ii) $x = x'$ or, $x \in \mathbf{C}(R, B_{R,R'})$ and $x' \in \mathbf{C}(R', B_{R,R'})$.*

When two states are adjacent, the *pairwise condition* in Proposition 1 reduces to that given in Proposition 2. While the pairwise condition is more complex than the condition given in Proposition 2, it is somewhat easier to interpret.

For globally distinct states, pairwise implementation is virtually unrestricted. It is sufficient that the outcomes are drawn from the Condorcet sets.[4] Since agenda voting always draws from the Condorcet set, this requirement is more generally necessary for implementation (see e.g. Moulin [1986], Lemma 9).

For globally similar states, there are stronger restrictions on what can be implemented. As is more generally necessary for implementation in this environment, the outcomes must be drawn from the same locale g (see e.g. Moulin [1986], Lemma 10). Within any given locale however, the restrictions are relatively weak. Any pair of outcomes that can be implemented on a subset g^* of g can also be implemented on X.

The proof of the main result follows directly from Proposition 1. The result is obtained by algebraic methods and does not rely on the explicit construction of a mechanism (i.e. an agenda). Since the approach is somewhat unconventional, it will be helpful to provide a brief overview.

To get the basic intuition, first consider any collection $\mathcal{R}^d$ of globally distinct states. Now, fix a pair of states $R_j, R_k \in \mathcal{R}^d$ and a pair of outcomes $x_j \in \mathbf{C}(X, R_j)$ and $x_k \in \mathbf{C}(X, R_k)$. From Proposition 1(I), there exists an agenda $T(j,k)$ such that $c_{T(j,k)}(R_j) = x_j$ and $c_{T(j,k)}(R_k) = x_k$. Let $\mathcal{T}^d$ define any collection of agendas $T(j,k)$ ranging over all pairs of outcomes $x_j \in \mathbf{C}(X, R_j)$, $x_k \in \mathbf{C}(X, R_k)$ and all pairs of states $R_j, R_k \in \mathcal{R}^d$. Let $\mathcal{C}(\mathcal{T}^d)$ define the collection of choice functions c_T that correspond to some $T \in \mathcal{T}^d$. Any agendas T_1 and T_2 in $\mathcal{T}^d$ may be joined at the root to obtain a new agenda $T_1 + T_2$ and a new agenda choice function $c_{T_1+T_2}$ such that

$$c_{T_1+T_2}(R) = \max_R \{c_{T_1}(R), c_{T_2}(R)\}$$

Applying a theorem in universal algebra due to Maroti [2002], it can be shown that the closure of $\mathcal{C}(\mathcal{T}^d)$ under agenda concatenation coincides with the collection of agenda choice functions that select from the Condorcet set in every state $R \in \mathcal{R}^d$. Formally:

Proposition 3 (Globally Distinct States) *For any collection of globally distinct states $\mathcal{R}^d$, the (partial) Condorcet social choice function $f^d : \mathcal{R}^d \to X$ is implementable iff $f^d(R) \in \mathbf{C}(X, R)$ for all $R \in \mathcal{R}^d$.*

In other words, Proposition 3 shows that the condition in Proposition 1(I) is necessary and sufficient for globally distinct states. The next result shows the necessity and sufficiency of the condition in Proposition 1(II) for globally similar states.

Proposition 4 (Globally Similar States) *For any complete collection of globally similar states $\mathcal{R}^s$, the (partial) Condorcet social choice function $f^s : \mathcal{R}^s \to X$ is implementable iff it is pairwise implementable for every pair of states $R, R' \in \mathcal{R}^s$.*

Like Proposition 3, the proof of this result leverages the algebraic structure of agendas. To get the basic intuition, consider any collection $\mathcal{R}^s$ of globally similar states with Condorcet set $\mathbf{C}$. Given an outcome $x \in \mathbf{C}$, it is not difficult to construct an agenda $T(x)$ that implements x

[4] Srivastava and Trick [1996] show the sufficiency of this condition when the Condorcet sets of R and R' are distinct (see Theorem 2 of their paper). It is easy to see that this is a corollary of Proposition 2.

for every $R \in \mathcal{R}^s$. To see this, suppose that $\mathbf{G}(X) = \{g_i\}_{i=1}^k$ is the maximal decomposition $\mathbf{C}$ so that $g_i R_G g_{i+1}$ for $i < k$ and $g_k R_G g_1$. To implement $x \in g_1$, construct an elimination agenda $T(x)$ using the list

$$L = (x, g_2, ..., g_k, g_1 \backslash \{x\}, X \backslash \mathbf{C})$$

At every node, append an agenda that contains every outcome in g_i (resp. $g_1 \backslash \{x\}$ and $X \backslash \mathbf{C}$). The fact that $c_{T(x)}(R) = x$ for every state $R \in \mathcal{R}^s$ is a simple extension of a result due to Miller [1977]. Define $\mathcal{T}^s = \{T(x) : x \in \mathbf{C}\}$ so that $\mathcal{C}(\mathcal{T}^s)$ describes a collection of agenda choice functions c_T that pick the same outcome for every $R \in \mathcal{R}^s$. Clearly, the collection $\mathcal{T}^s$ satisfies the *pairwise condition*. Proposition 4 then follows by establishing that the closure of $\mathcal{C}(\mathcal{T}^s)$ under agenda concatenation generates *all* of the social choice functions that satisfy the *pairwise condition*. Given Propositions 3 and 4, the main result then follows by induction on the number of globally distinct sub-collections in $\mathcal{R}$. The details are presented in Section 4.

4 Proofs

4.1 Proof of Proposition 1

Proposition 1(I) follows from Theorem 1 of Srivastava and Trick [1996]. The following definition is required to state this result: a subset $PS \subseteq X$ is *prime* if there is no non-trivial partition $\mathbf{PS} = \{PS_i\}_{i=1}^k$ of PS such that: (i) $\mathbf{PS}$ is a decomposition of R and R' on PS; and, (ii) the quotient relations induced by $\mathbf{PS}$ agree so that $R/\mathbf{PS} = R'/\mathbf{PS}$.

Theorem 1 of Srivastava and Trick *The outcomes x and x' are pairwise implementable on states R and R' for some subset of X iff there exists a prime set PS such that $\{x, x'\} \subseteq PS \subseteq X$, $x \in \mathbf{C}(PS, R)$, and $x' \in \mathbf{C}(PS, R')$.*

Proposition 1(I) is a consequence of the following lemma.

Lemma 1 *If $\langle \mathbf{G}(X), R_G \rangle \neq \langle \mathbf{G}'(X), R'_G \rangle$, $\mathbf{C}(X, R) \cup \mathbf{C}(X, R')$ is a prime set.*

Proof. Omitted due to lack of space. ■

Proof of Proposition 1(I). ($\Leftarrow$) By Lemma 1, $\mathbf{C} \cup \mathbf{C}'$ is a prime set. By Theorem 1 of Srivastava and Trick, any $x \in \mathbf{C} = \mathbf{C}(\mathbf{C} \cup \mathbf{C}', R)$ and $x' \in \mathbf{C}' = \mathbf{C}(\mathbf{C} \cup \mathbf{C}', R')$ are pairwise implementable for $\mathbf{C} \cup \mathbf{C}' \subseteq X$. To complete the proof, fix a pair $x \in \mathbf{C}$ and $x' \in \mathbf{C}'$ and an agenda T that implements (x, x') for $\mathbf{C} \cup \mathbf{C}'$. Next, construct an agenda whose left branch at the root corresponds with T and whose right branch is any agenda on $X \backslash (\mathbf{C} \cup \mathbf{C}')$. (When $X \backslash (\mathbf{C} \cup \mathbf{C}') = \emptyset$, the right branch can be omitted.) By construction, the desired outcome emerges from the left branch in each state and defeats whatever emerges from the right. ($\Rightarrow$) If x and x' are pairwise implementable, $x \in \mathbf{C}$ and $x' \in \mathbf{C}'$ (by Lemma 9 of Moulin [1986]). ■

Proposition 1(II) is a consequence of the following lemma:

Lemma 2 *Given a collection of globally similar states $\mathcal{R}^s$, the (partial) Condorcet social choice function $f^s : \mathcal{R}^s \to X$ is implementable iff f^s is implementable for some $g^* \subseteq g \in \mathbf{G}(X)$.*

Proof. For parsimony, let $\mathbf{C} = \mathbf{C}(X, R)$ and $\mathbf{G}(X) = \{g_i\}_{i=1}^k$. First, fix an element $x \in g_i$ and suppose that $g_i R_G g_{i+1}$ for $i < k$ and $g_k R_G g_1$ (otherwise, the components can be relabeled). Construct an elimination agenda $T(x)$ using $L = (x, ..., g_k, g_1 \backslash g\{x\}, X \backslash \mathbf{C})$. As in the proof of

Proposition 1(I), the bottom branch may be omitted when $X \backslash \mathbf{C} = \emptyset$. To the branch labelled x, append the item x. To the branches labelled by g_i (respectively $g_1 \backslash \{x\}$ or $X \backslash \mathbf{C}$), append an agenda T_i containing every outcome in g_i (resp. $g_1 \backslash \{x\}$ or $X \backslash \mathbf{C}$). By construction, $T(x)$ implements x on $\mathcal{R}^s$ (see e.g. Lemma 8.3.3 of Laslier [1997]). Moreover, it can be associated with the trivial agenda $t(x) = x$ that implements x on $\{x\} \subseteq g_1$.

Let $\mathcal{T}_1 = \{T(x) : x \in \mathbf{C}\}$ define the collection of agendas $T(x)$ on $\mathbf{C}$. Similarly, let $\mathcal{T}_1(g^*) = \{t(x) : x \in g^*\}$ define the collection of agendas $t(x)$ on $g^* \subseteq g \in \mathbf{G}(X)$. By construction, any agenda-implementable f^s on X can be obtained by concatenating agendas in $\mathcal{T}_1$. Since $f^s(R) \in \mathbf{C}$ for all $R \in \mathcal{R}^s$ (by Lemma 9 of Moulin [1986]), one can ignore agendas $T(x)$ where $x \notin \mathbf{C}$. Likewise, any agenda-implementable f^s on $g^* \subseteq g$ can be obtained by concatenating agendas in $\mathcal{T}_1(g^*) = \{x : x \in g^*\}$.

Define $\mathcal{T}_n = \{T_{n-1} + T_k : T_{n-1} \in \mathcal{T}_{n-1}$ and $T_k \in \mathcal{T}_k$ for $k < n\}$ and let $\mathcal{C}_n = \{f^s : f^s = c(T_n)$ for some $T_n \in \mathcal{T}_n\}$ (where $c(T_n)$ is the Condorcet social choice function implemented by T_n). Likewise, let $\mathcal{T}_n(g^*) = \{t_{n-1}(g^*) + t_k(g^*) : t_{n-1}(g^*) \in \mathcal{T}_{n-1}(g^*)$ and $t_k(g^*) \in \mathcal{T}_k(g^*)$ for $k < n\}$ and let $\mathcal{C}_n(g^*) = \{f^s : f^s = c(t_n)$ for $t_n \in \mathcal{T}_n(g^*)\}$.

Using strong induction, I establish: $f^s = c(T_n) \in \mathcal{C}_n$ iff $f^s = c(t_n(g^*)) \in \mathcal{C}_n(g^*)$ for some $g^* \subseteq g \in \mathbf{G}(X)$. The claim is trivial for the base case $n = 1$. So, suppose it holds for $n \leq N$.

($\Rightarrow$) Now, consider any $f^s = c(T_N + T_k) \in \mathcal{C}_{N+1}(\mathcal{T})$. By the induction step, $c(T_N) = c(t_N(g_1^*))$ for some $t_N(g_1^*)$ on $g_1^* \subseteq g_1$ and $c(T_k) = c(t_k(g_2^*))$ for some $t_k(g_2^*)$ on $g_2^* \subseteq g_2$. There are two cases: (i) $g_1 \neq g_2$; and, (ii) $g_1 = g_2$. (i) Suppose, without loss of generality, that $g_1(R_G)g_2$. Then:

$$f^s = c(T_N + T_k) = c(T_N) + c(T_k) = c(T_N) = c(t_N(g_1^*))$$

where $t_N(g_1^*)$ implements f^s on $g_1^* \subseteq g_1 \in \mathbf{G}(X)$ (by the induction step). (ii) In this case:

$$f^s = c(T_N + T_k) = c(T_N) + c(T_k) = c(t_N(g_1^*)) + c(t_k(g_2^*)) = c(t_N(g_1^*) + t_k(g_2^*))$$

where $t_N(g_1^*) + t_k(g_2^*)$ implements f^s on $g_1^* \cup g_2^* \subseteq g_1 \in \mathbf{G}(X)$.

($\Leftarrow$) Suppose $f^s = c(t_N(g_1^*) + t_k(g_2^*)) \in \mathcal{C}_{N+1}(g^*)$ for $t_N(g_1^*)$ on $g_1^* \subseteq g^* \subseteq g$ and $t_k(g_2^*)$ on $g_2^* \subseteq g^* \subseteq g$. By the induction step, $c(t_N(g_1^*)) = c(T_N)$ and $c(t_k(g_2^*)) = c(T_k)$ for $c(T_N) \in \mathcal{C}_N$ and $c(T_k) \in \mathcal{C}_k$. Following the same reasoning as case (ii) above, $f^s = ... = c(T_N + T_k)$ where $T_N + T_k$ implements f^s. ■

Proof of Proposition 1(II). Given Lemma 2, let $\mathcal{R}^s = \{R, R'\}$. ■

4.2 Proofs of Proposition 3, Proposition 4, and the Main Result

The proofs of these results rely on algebraic methods. Some preliminary definitions are required.

4.2.1 Preliminaries

Given a pairwise-ranking R on X, let the *tournament algebra* $\mathbf{X}$ be defined by a pair $(X, +)$ consisting of X and a binary operation $+$ such that $x + y = x$ iff xRy or $x = y$.[5] In turn, tournament algebras can be extended to products. Given a collection $\{\mathbf{X}_i\}_{i=1}^m$ of tournament algebras, the *product algebra* $\Pi_{i=1}^m \mathbf{X}_i$ is defined by $(\Pi_{i=1}^m X_i, +)$ where $+$ applies the operations $+_i$ component-wise so that $x + y \equiv (x_i +_i y_i)_{i=1}^m$. The projection of $x \equiv (x_i)_{i=1}^m \in \Pi_{i=1}^m X_i$ onto any collection $J \subseteq \{1, ..., m\}$ of components is $\pi_J(x) = \Pi_{i \in J} x_i$. A *subdirect product* of $(\Pi_{i=1}^m X_i, +)$ is a sub-algebra $\mathbf{Y} \equiv (Y, +)$ of $\Pi_{i=1}^m \mathbf{X}_i$ (i.e. $Y \subseteq \Pi_{i=1}^m X_i$ and Y is closed under the

[5] More generally, an algebra $\mathbf{X}$ is a set X that is algebraically closed under a collection of n-ary operations.

binary operation $+$) such that $Y_i \equiv \{\pi_{\{i\}}(y) : y \in Y\} = X_i$ for any component Y_i. The subdirect product $\mathbf{Y}$ is *weakly indecomposable* if there exists no bi-partition (J, K) of the m components such that $Y = \pi_J(Y) \times \pi_K(Y)$ (up to re-ordering of the components).

A tournament algebra $(X, +)$ is *cyclic* if $\mathbf{C}(X, R) = X$ where R is the relation induced by the binary operation $+$ (so that xRy iff $x + y = x$ and $x \neq y$). A *congruence* β on $\mathbf{Y} \equiv (Y, +)$ is an equivalence relation on Y such that $(x + y)\beta(x' + y')$ iff $x\beta x'$ and $y\beta y'$. The largest congruence on $\mathbf{Y}$ is the *complete relation* $\mathbf{1}_Y = Y \times Y$ while the smallest is the *trivial relation* $\mathbf{Id}_Y = \{(y, y) : y \in Y\}$. Given a congruence β on $\mathbf{Y}$, the *quotient algebra* $\mathbf{Y}/\beta$ is $(Y/\beta, +_\beta)$ where Y/β is the partition of Y induced by β and $+_\beta$ is the binary operation $y/\beta +_\beta y'/\beta \equiv \{Z \in Y/\beta : y + y' \in Z\}$. Finally, $\mathbf{Y}$ is *irreducible* when its only congruences are $\mathbf{1}_Y$ and $\mathbf{Id}_Y$.

4.2.2 Proofs

The proofs of these results rely on a theorem in universal algebra established by Maroti [2002] (combining Lemmas 5.10 and 5.14 of his Ph.D. dissertation). To state Maroti's theorem:

Theorem (Maroti) *Let Y be a weakly indecomposable subdirect product of m cyclic tournament algebras. Then, Y has a unique largest congruence $\beta \neq Y \times Y$ and Y/β is an irreducible tournament algebra.*

They also rely on the following:

Claim 1 ***(I)*** *Natural numbers h and $h + 1$ are co-prime.* ***(II)*** *If a and b are co-prime, then every pair of congruence relations of the form $x = k(\operatorname{mod} a)$ and $x = l(\operatorname{mod} b)$ has a solution.*

Proof. Omitted due to lack of space. ■

To simplify the presentation below, consider the following definitions. Let $\mathcal{R}(X) = \{R_i\}_{i \in I}$ denote the collection of states on X. For parsimony, I abbreviate $\mathbf{C}(X, R_i)$ to $\mathbf{C}_i$. If there are n outcomes, denote the domain by X_n so that $\mathcal{R}(n)$ defines the collection of states on X_n. Let $\mathcal{R}^d_J(X) = \{R_j\}_{j \in J}$ denote a collection of $J \subseteq I$ globally distinct states in $\mathcal{R}(X)$ so that $\mathcal{R}^d(n)$ denotes any maximal collection of globally distinct states in $\mathcal{R}(n)$. Let $\mathcal{R}^s_j(n)$ denote the maximal collection (or *class*) of states that are globally similar to $R_j \in \mathcal{R}^d(n)$ and let $K(j) \subseteq I$ denote the set of indices associated with $\mathcal{R}^s_j(n)$. Finally, let $\mathbf{R}(n) = \{\mathcal{R}^s_j(n)\}_{j \in J}$ denote the partition dividing $\mathcal{R}(n)$ into classes of globally similar states.

It is possible to identify any Condorcet social choice function $c : \mathcal{R}(X) \to X$ with a vector $\vec{x} \equiv (x_i)_{i \in I} \in \Pi_{i \in I} X$. Using this approach, let $\mathcal{C}(n) = \{\vec{x} \in \Pi_{i \in I}\mathbf{C}_i : \vec{x} \text{ is implementable}\}$ denote the collection of agenda-implementable Condorcet social choice functions on X_n. Let $\mathcal{C}^d_J(X) = \{\pi_J(\vec{x}) \in \Pi_{j \in J}\mathbf{C}_j : \vec{x} \in \mathcal{C}(X)\}$ denote the collection of agenda-implementable Condorcet social choice functions on $\mathcal{R}^d_J(X)$. And, let $\mathcal{C}^s_j(n) = \{\pi_{K(j)}(\vec{x}) \in \Pi_{k \in K(j)}\mathbf{C}_k : \vec{x} \in \mathcal{C}(n)\}$ denote the collection of agenda-implementable Condorcet social choice functions on $\mathcal{R}^s_j(n) = \{R_k\}_{k \in K(j)}$.

Proof of Proposition 3. ($\Rightarrow$) If $f^d : \mathcal{R}^d \to X$ is implementable, it is also pairwise implementable for every pair of states in $\mathcal{R}^d$. From Proposition 1(I), $f^d(R) \in \mathbf{C}$ for all $R \in \mathcal{R}^d$.

($\Leftarrow$) Let $\mathcal{R}^d_I = \{R_i\}_{i \in I}$ and suppose that $|\mathbf{C}_i| > 1$. To establish the result, I show $\mathcal{C}^d_I(X) = \Pi^I_{i=1}\mathbf{C}_i$. The proof is by induction on the number of globally distinct states I. Proposition 1(I) proves the base case $I = 2$. Assume that the result holds for $|I| = n$. To complete the induction, I show the result $|I| = n + 1$. To simplify the notation, let $\bar{X} \equiv \Pi^{n+1}_{i=1}\mathbf{C}_i$ and $Y \equiv \mathcal{C}_{n+1}(\mathcal{T}^d)$ so that $\bar{X}_J = \pi_J(\bar{X})$ and $Y_J = \pi_J(Y)$ define the projections onto the sub-collection of states in J. To establish $Y = \bar{X}$, suppose otherwise.

First, note that Y is a subdirect product of $\bar{X}$. By the induction hypothesis, $\mathcal{C}^d_{J(n)}(X) = \Pi_{i \in J(n)} \mathbf{C}_i$ for any collection $J(n)$ of n states. Accordingly, $\pi_i(\mathcal{C}^d_{J(n)}(X)) = \mathbf{C}_i$. Second, each component of Y is cyclic because $Y_i = \mathbf{C}_i$. Finally, Y is weakly indecomposable. To see this, suppose $Y = \pi_J(Y) \times \pi_K(Y)$. By the induction step, $\pi_J(Y) = \Pi_{j \in J} \mathbf{C}_j$ and $\pi_K(Y) = \Pi_{k \in K} \mathbf{C}_k$ so that $Y = \Pi_{j \in J} \mathbf{C}_j \times \Pi_{k \in K} \mathbf{C}_k = \bar{X}$. But this contradicts the assumption that $Y \neq \bar{X}$ and establishes Y is weakly indecomposable. As such, Maroti's theorem applies. Let β define the largest congruence of Y such that $\beta \neq Y \times Y$. There are two cases to consider: (i) $|\bar{X}_j| = |\bar{X}_k| = h + 1 > 1$ for all $j, k \leq n + 1$; and (ii) there are distinct states j and k such that $|\bar{X}_j| \neq |\bar{X}_k|$.

(i) Pick any two states j and k and consider any distinct a, $b \in Y$. Label the elements of $\bar{X}_j$ so that the sequence $\{x^l_j\}^{h+1}_{l=0}$ defines a complete cycle $x^0_j R_j ... R_j x^l_j R_j ... R_j x^{h+1}_j = x^0_j$ in $\bar{X}_j$. And, label the elements of $\bar{X}_k$ so that $\{x^m_k\}^{h+1}_{m=0}$ defines a complete *reverse cycle* $x^0_k = x^{h+1}_k R_k ... R_k x^m_k R_k ... R_k x^0_k$ in $\bar{X}_k$. By the base case, there is a $x^{(l,m)}_{-jk} \in \Pi_{i \in I \setminus \{j,k\}} X_i$ s.t. $x^{(l,m)} \equiv (x^l_j \times x^m_k \times x^{(l,m)}_{-jk}) \in Y$. Without loss of generality, let $a \equiv x^{(0,0)}$. By construction, $x^{(l,m)}$ and $x^{(l+1,m+1)}$ are unranked by $\Pi^{N+1}_{i=1} R_i$. Since Y/β is a tournament, $(x^{(l,m)}, x^{(l+1,m+1)}) \in \beta$ for $l \leq h$ and $m \leq h$ so that $(a, x^{(l+1,m+1)}) \in \beta$.

By Theorem 7 of Harary and Moser [1966], there exists an h-length cycle $C_j \subseteq X_j$ containing b_j. Let l^* be the lowest index l such that $x^l_j \in C_j$ and let $x^* = x^{(l^*,l^*)}$. So, it is possible to label the elements of C_j so that the sequence $\{x^l_j\}^h_{l=0}$ defines a complete cycle $x^{l^*}_j = x^0_j R_j ... R_j x^l_j R_j ... R_j x^h_j = x^0_j$ in C_j. Because h and $h + 1$ are co-prime, $(x^{(l,m)}, x^{(l',m')}) \in \beta$ for any l, $l' \leq h$ and m, $m' \leq h + 1$ (by Claim 1). In particular, $(x^*, b) \in \beta$. Since $(a, x^*) \in \beta$ (by the first argument), it then follows that $(a, b) \in \beta$ so that $\beta = Y \times Y$.

(ii) Fix components j and k such that $|\bar{X}_j| = h' > h = |\bar{X}_k|$ and consider any distinct a, $b \in Y$. By the same approach as in the previous case, define a complete cycle on $\bar{X}_j$ and a complete reverse cycle on $\bar{X}_k$ such that a corresponds to the first element in each sequence. By Theorem 7 of Harary and Moser [1966], there exists an $(h+1)$-length cycle $C_j \subseteq X_j$ that contains b_j. Let l^* be the lowest index l such that $x^l_j \in C_j$ and let $x^* = x^{(l^*,l^*)}$. By the same argument given in the previous case, $(a, x^*) \in \beta$ and $(x^*, b) \in \beta$ so that $(a, b) \in \beta$ so that $\beta = Y \times Y$.

In both cases, $\beta = Y \times Y$ follows from $Y \neq \bar{X}$. But this contradicts $\beta \neq Y \times Y$. Thus, $Y = \bar{X}$. Given any collection of distinct states $\mathcal{R}^d$, it then follows that f^d is implementable if $f^d(R_j) \in \mathbf{C}_j$ for all $R_j \in \mathcal{R}^d$. The proof covers $\mathcal{R}^d$ consisting of *non-trivial* states such that $|\mathbf{C}_j| > 1$. This is sufficient to establish the result for any collection of distinct states $\mathcal{R}^d$. ∎

The following lemma is needed in the proof of Proposition 4:

Lemma 3 *Given a complete collection of globally similar states $\mathcal{R}^s$, the (partial) Condorcet social choice function $f^s : \mathcal{R}^s \to X$ is implementable for every pair of states in $\mathcal{R}^s$ iff f^s is implementable for every pair of states on a subset g^* of some $g \in \mathbf{G}(X)$.*

Proof. Let $\mathcal{PW}(n) = \{\vec{x} \in \Pi_{i \in I} \mathbf{C}_i : \vec{x}$ satisfies the *pairwise condition* on $\mathcal{R}(n)\}$ represent the collection of Condorcet social choice functions that are pairwise implementable on X_n. Now, consider the similarity class $\mathcal{R}^s(n) = \{R_k\}_{k \in K}$ with global structure $\mathbf{G}(X_n) = \{g_l\}_{l \in L}$. Let $\mathcal{PW}^s(n) = \{\pi_K(\vec{x}) \in \Pi_{k \in K} \mathbf{C}_k : \vec{x} \in \mathcal{PW}(n)\}$ represent the choice functions that are pairwise implementable on $\mathcal{R}^s(n)$. First note that:

$$\mathcal{PW}^s(n) = \bigcup_{l \in L} \mathcal{PW}^s_l(n)$$

where $\mathcal{PW}^s_l(n) = \{\pi_K(\vec{x}) \in \Pi_{k \in K} \mathbf{C}_k : \vec{x} \in \mathcal{PW}^s(n) \cap \Pi_{k \in K} g_l\}$ is the sub-collection of $\mathcal{PW}^s(n)$ selecting from $g_l \in \mathbf{G}(X_n)$. To see this, fix adjacent states R and R' in $\mathcal{R}^s(n)$ such that

$f^s(R) = x \in g_l$ and $f^s(R') = x'$. By assumption, f^s is pairwise implementable for R and R'. From Proposition 1(II), $x \in g_l$ implies $x' \in g_l$. By the same argument, $f^s(R'') \in g_l$ for all $R'' \in \mathcal{R}^s(n)$.

Let $\mathcal{PW}_l^s(n)|_{g^*}$ define the sub-collection of $\mathcal{PW}_l^s(n)$ that is pairwise implementable on $g^* \subseteq g_l$. And, let $\mathcal{PW}_l^s(n)[g^*]$ define the sub-collection of $\mathcal{PW}_l^s(n)$ with range $g^* \subseteq g_l$ (so that $\cup_{k\in K}\{f^s(R_k)\} = g^*$ for any $f^s \in \mathcal{PW}_l^s(n)[g^*]$). By construction, $\mathcal{PW}_l^s(n) = \bigcup_{g^*\subseteq g_l} \mathcal{PW}_l^s(n)[g^*]$. To establish the desired result, it suffices to prove $\mathcal{PW}_l^s(n)|_{g^*} = \mathcal{PW}_l^s(n)[g^*]$ for any $g^* \subseteq g_l$. Using this identity, it follows that

$$\mathcal{PW}^s(n) = \bigcup_{l\in L} \bigcup_{g^*\subseteq g_l} \mathcal{PW}_l^s(n)|_{g^*}$$

as required. To show $\mathcal{PW}_l^s(n)|_{g^*} = \mathcal{PW}_l^s(n)[g^*]$ for any $g^* \subseteq g_l$, first consider the following:

Claim A *If $f^s \in \mathcal{PW}_l^s(n)$, $\{R, R'\} \subseteq \mathcal{R}^s(n)$, and $\mathbf{C}(g_l, R) = \{f^s(R')\}$, then $f^s(R) = f^s(R')$.*

Proof. Omitted due to lack of space. ■

The result follows by establishing that $\mathcal{PW}_l^s(n)[g^*] \subseteq \mathcal{PW}_l^s(n)|_{g^*}$. The inverse inclusion $\mathcal{PW}_l^s(n)|_{g^*} \subseteq \mathcal{PW}_l^s(n)[g^*]$ follows from the fact that $f^s(R) = x$ for any $R \in \mathcal{R}^s(n)$ such that xRx' for all $x' \in g^*\backslash\{x\}$ (by Lemma 9 of Moulin [1986]). To establish $\mathcal{PW}_l^s(n)[g^*] \subseteq \mathcal{PW}_l^s(n)|_{g^*}$, there are two cases to consider: (i) $g^* = g_l$; and, (ii) $g^* \subsetneq g_l$.

(i) For globally distinct states such that $\mathbf{G}(g_l, R) \neq \mathbf{G}(g_l, R')$, it is sufficient to show that $f^s(R) \in \mathbf{C}(g_l, R)$ for all $R \in \mathcal{R}^s(n)$. To see this, consider $f^s \in \mathcal{PW}_l^s(n)[g_l]$ and fix some state R such that $|\mathbf{C}(g_l, R)| > 1$ and any $x' \in \mathbf{C}(g_l, R)$. (The fact that $f^s(R) \in \mathbf{C}(g_l, R)$ for any R such that $|\mathbf{C}(g_l, R)| = 1$ follows from Claim A and the assumption that $f^s \in \mathcal{PW}_l^s(n)[g_l]$.) Consider the state R' such that $R'|_{X\backslash g_l} = R|_{X\backslash g_l}$, $R'|_{g_l\backslash\{x'\}} = R|_{g_l\backslash\{x'\}}$, and $x'R'x$ for any $x \in g_l\backslash\{x'\}$. Since $f^s \in \mathcal{PW}_l^s(n)[g_l]$, x' is chosen for some state $R'' \in \mathcal{R}^s(n)$. By Claim A, it follows that $f^s(R') = x'$. By construction, $\{x'\} \subseteq PS \subseteq \mathbf{C}(g_l, R)$ for any non-trivial prime set PS on R and R'. By Theorem 1 of Srivastava and Trick, it then follows that $f^s(R) \in \mathbf{C}(g_l, R)$. This establishes $f^s(R) \in \mathbf{C}(g_l, R)$ for all $R \in \mathcal{R}^s(n)$.

Next, consider globally similar states such that $\mathbf{G}(g_l, R) = \mathbf{G}(g_l, R') = \{g_l^i\}_{i\in I}$. Without loss of generality, suppose $f^s(R) \in g_l^i$. It is sufficient to show that $f^s(R)$ and $f^s(R')$ are pairwise implementable for some $g \subseteq g_l^i$. From Theorem 1 of Srivastava and Trick, $f^s(R)$ and $f^s(R')$ are pairwise implementable for some prime set PS such that $f^s(R) \in PS$. By definition, it must be that $PS \subseteq g_l^i$ for any prime set such that $f^s(R) \in PS$. This establishes the desired result.

(ii) Pick $f^s \in \mathcal{PW}_l^s(n)[g^*]$ for some $g^* \subsetneq g_l$. Fix a state R and consider the state $R^{\downarrow g^*}$ defined by $R^{\downarrow g^*}|_{X\backslash g^*} = R|_{X\backslash g^*}$, $R^{\downarrow g^*}|_{g^*} = R|_{g^*}$, and $x'R^{\downarrow g^*}x$ for any $x' \in X\backslash g^*$ and any $x \in g^*$. By construction, any non-trivial prime set PS on R and $R^{\downarrow g^*}$ must contain some $x' \in X\backslash g^*$. Since $f^s(R)$ and $f^s(R^{\downarrow g^*})$ are pairwise implementable, $f^s(R) = f^s(R^{\downarrow g^*})$. Otherwise, $\mathbf{C}(PS, R^{\downarrow g^*}) \subseteq X\backslash g^*$ so that $f^s(R^{\downarrow g^*}) \in X\backslash g^*$ which contradicts the assumption that $f^s \in \mathcal{PW}_l^s(n)[g^*]$. This establishes that $f^s(R) = f^s(R')$ for any states R and R' in $\mathcal{R}^s(n)$ such that $R|_{g^*} = R'|_{g^*}$.

To see that $f^s(R) \in \mathbf{C}(g^*, R|_{g^*})$ for any $R \in \mathcal{R}_j^s(n)$, fix a state $\bar{R}$ such that $x\bar{R}x'$ for any $x \in g^*$ and $x' \in g_l\backslash g^*$. By the same reasoning as in (i) above, $f^s(\bar{R}) \in \mathbf{C}(g^*, \bar{R})$. Since $f^s(R) = f^s(\bar{R})$ for any R and $\bar{R}$ in $\mathcal{R}^s(n)$ such that $R|_{g^*} = \bar{R}|_{g^*}$, then $f^s(R) \in \mathbf{C}(g^*, R)$ for all $R \in \mathcal{R}^s(n)$.

To complete the proof, fix any state R and consider the state $R^{\uparrow g^*}$ defined by $R^{\uparrow g^*}|_{X\backslash g^*} = R|_{X\backslash g^*}$, $R^{\uparrow g^*}|_{g^*} = R|_{g^*}$, and $xR^{\uparrow g^*}x'$ for any $x \in g^*$ and $x' \in X\backslash g^*$. Now consider any R' globally similar to R on g^*. By construction, $R^{\uparrow g^*}|_{g^*} = R|_{g^*}$ and $R'^{\uparrow g^*}|_{g^*} = R'|_{g^*}$ so that $f^s(R) = f^s(R^{\uparrow g^*})$ and $f^s(R') = f^s(R'^{\uparrow g^*})$. Moreover, $R^{\uparrow g^*}$ and $R'^{\uparrow g^*}$ are globally similar on g^*. Without loss of generality, suppose that $\mathbf{G}(g^*, R) = \mathbf{G}(g^*, R') = \{g_i^*\}_{i\in I}$ and $f^s(R) \in g_i^*$.

Following the same reasoning as in (i) above, $f^s(R^{\uparrow g^*})$ and $f^s(R'^{\uparrow g^*})$ are pairwise implementable for some prime set $PS \subseteq g_i^*$, which establishes the desired result. ■

Proof of Proposition 4 and Main Result. (⇒) If $f : \mathcal{R} \to X$ (respectively $f^s : \mathcal{R}^s \to X$) is implementable, it is implementable for every pair of states in $\mathcal{R}$ (respectively $\mathcal{R}^s$).

(⇐) As in Lemma 3, let $\mathcal{PW}(n)$ represent the choice functions that satisfy the pairwise condition on $\mathcal{R}(n)$ and let $\mathcal{PW}_j^s(n)$ represent the choice functions that satisfy the pairwise condition on the similarity class $\mathcal{R}_j^s(n) = \{R_k\}_{k \in K(j)}$. Finally, let $J(n) = |\mathbf{R}(n)|$ represent the number of similarity classes in $\mathcal{R}(n)$. For Proposition 4, I show **(I)** $\mathcal{C}_j^s(n) = \mathcal{PW}_j^s(n)$ for any $j \in J(n)$. For the main result, I show **(II)** $\mathcal{C}(n) = \Pi_{j \in J} \mathcal{C}_j^s(n)$ for any n. Results **(I)** and **(II)** establish $\mathcal{C}(n) = \Pi_{j \in J} \mathcal{PW}_j^s(n)$. Since $\mathcal{PW}(n) = \Pi_{j \in J} \mathcal{PW}_j^s(n)$ by Proposition 1, it follows that $\mathcal{C}(n) = \mathcal{PW}(n)$. The proof is by strong induction on the size of the domain n and the number of similarity classes $J(n)$.

For $n \in \{1, 2, 3\}$, it is easy to see that **(I)** and **(II)** hold. (For $n = 2$, there are 2 globally distinct states each consisting of a linear order. For $n = 3$, there are 8 states and 5 similarity classes (3 classes that consist of two linear orders each and 2 classes consisting of one cycle).

For all $m < n$, assume $\mathcal{C}(m) = \Pi_{j \in J(m)} \mathcal{C}_j^s(m)$ and $\mathcal{C}_j^s(m) = \mathcal{PW}_j^s(m)$ for all $j \in J(m)$. In order to complete the induction, it is enough to show that **(I)** and **(II)** hold for n.

(I) Consider any non-trivial class similarity $\mathcal{R}_j^s(n) \in \mathbf{R}(n)$ (so that $|\mathcal{R}_j^s(n)| > 1$ or, equivalently, $|\mathbf{G}^j(X_n)| > 1$). Wlog, suppose $\mathbf{G}^j(X) = \{g_l^j\}_{l \in L(j)}$ so that $|g_l^j| < n$. By Lemma 2:

$$\mathcal{C}_j^s(n) = \bigcup_{l \in L(j)} \bigcup_{g^* \subseteq g_l^j} \mathcal{C}_{jl}^s(n)|_{g^*}$$

where $\mathcal{C}_{jl}^s(n)|_{g^*}$ is the collection of Condorcet social choice functions that are implementable on $g^* \subseteq g_l^j$. Lemma 3 above establishes that:

$$\mathcal{PW}_j^s(n) = \bigcup_{l \in L(j)} \bigcup_{g^* \subseteq g_l^j} \mathcal{PW}_{jl}^s(n)|_{g^*}$$

By induction assumptions **(I)** and **(II)**, $\mathcal{C}_{jl}^s(n)|_{g^*} = \mathcal{PW}_{jl}^s(n)|_{g^*}$ for any $g^* \subseteq g_l^j$. Consequently, $\mathcal{C}_j^s(n) = \mathcal{PW}_j^s(n)$ which establishes the desired result.

(II) First, let $J^*(n) = \{j \in J(n) : |\mathbf{C}_j| > 1\}$. Given $\mathcal{C}_j^s(n) = \mathcal{PW}_j^s(n)$ for every $j \in J^*(n)$, the result follows by induction on J. For ease of notation, let $\pi_J(\mathcal{C}(n)) = \pi_J$. To establish the base case $J = \{1, 2\}$, suppose $\pi_{\{1,2\}} \neq \pi_1 \times \pi_2$. Note that $\pi_{\{1,2\}}$ is a subdirect product of $\pi_1 \times \pi_2$. For any state $R_j \in \mathcal{R}_j^s(n)$, there exists an agenda $T(x)$ that implements every outcome in $x \in \mathbf{C}_j$. (The construction is similar to that given in Lemma 2.) This observation also establishes that the sub-algebra on each state is cyclic. Finally, the assumption that $\pi_{\{1,2\}} \neq \pi_1 \times \pi_2$ implies that $\pi_{\{1,2\}}$ is weakly indecomposable. To see this, suppose that there are two disjoint collections $\mathcal{R}_U = \{R_u : u \in U\}$ and $\mathcal{R}_V = \{R_v : v \in V\}$ such that $\mathcal{R}_U \cup \mathcal{R}_V = \mathcal{R}_1^s(n) \cup \mathcal{R}_2^s(n)$ and $\pi_{\{1,2\}} = \pi_U(\mathcal{C}(n)) \times \pi_V(\mathcal{C}(n))$. Now, consider any $R_1, R_1' \in \mathcal{R}_1^s(n)$ and suppose that $R_1 \in \mathcal{R}_U$ and $R_1' \in \mathcal{R}_V$. It follows that it is possible to pairwise implement $x \in g$ and $x' \in g'$ for $g \neq g'$. This contradicts Proposition 1 and establishes $\mathcal{R}_1^s(n) \subseteq \mathcal{R}_U$ or $\mathcal{R}_1^s(n) \subseteq \mathcal{R}_V$. A similar argument shows $\mathcal{R}_2^s(n) \subseteq \mathcal{R}_U$ or $\mathcal{R}_2^s(n) \subseteq \mathcal{R}_V$. Since the collections $\mathcal{R}_U$ and $\mathcal{R}_V$ are non-trivial, then $\mathcal{R}_1^s(n) = \mathcal{R}_U$ and $\mathcal{R}_1^s(n) = \mathcal{R}_V$ without loss of generality. But, this contradicts the assumption that $\pi_{\{1,2\}} \neq \pi_1 \times \pi_2$ and establishes that $\pi_{\{1,2\}}$ is weakly indecomposable.

Accordingly, the theorem of Maroti applies. Let β define the largest congruence of Y such that $\beta \neq \pi_{\{1,2\}} \times \pi_{\{1,2\}}$. By Proposition 1, it is possible to pairwise implement (x_1, x_2) and (x_1', x_2') on $R_1 \in \mathcal{R}_1^s(n)$ and $R_2 \in \mathcal{R}_2^s(n)$ so that $x_1 R_1 x_1'$ and $x_2' R_2 x_2$. Using the same approach as in Proposition 3, it follows that $\beta = \pi_{\{1,2\}} \times \pi_{\{1,2\}}$. But, this contradicts the assumption that $\beta \neq \pi_{\{1,2\}} \times \pi_{\{1,2\}}$ and establishes that $\pi_{\{1,2\}} = \pi_1 \times \pi_2$ in the base case $J = \{1,2\}$.

Now, assume that the result holds for $|J| = j$. In order to complete the induction, it suffices to show that the result holds for $|J| = j+1$. Following the same line of argument as in the base case (and Proposition 3), the result $\pi_J = \Pi_{j \in J} \pi_j$ can be established by contradiction. This proves $\pi_{J^*(n)}(\mathcal{C}(n)) = \Pi_{j \in J^*(n)} \mathcal{C}^s_j(n)$. It then follows that $\mathcal{C}(n) = \Pi_{j \in J(n)} \mathcal{C}^s_j(n)$. ■

References

[1] Abreu & Sen (1990), "Subgame Perfect Implementation", *J, Econ. Theory* 50 285-299.

[2] Black (1958), *The Theory of Committees and Elections*, New York: Cambridge University Press.

[3] Dutta & Sen (1993), "Implementing Generalized Condorcet Social Choice Functions via Backward Induction", *Soc. Choice Welfare* 10 149–160.

[4] Farquharson (1969), *Theory of Voting*, New Haven: Yale University Press

[5] Gol' berg & Gourvitch (1986), "Secret and Extensive Dominance Solvable Voting Schemes", Moscow *mimeo.*

[6] Harary & Moser (1966), "The Theory of Round Robin Tournaments", *Am. Math. Mon.* 73(3) 231–246.

[7] Herrero & Srivastava (1992), "Implementation via Backward Induction", *J. Econ. Theory* 56 70–88.

[8] Laslier (1997), *Tournament Solutions and Majority Voting*, Berlin/Heidelburg: Springer Verlag.

[9] Maroti (2002), "The Variety Generated by Tournaments", Vanderbilt University Ph.D. Dissertation.

[10] McKelvey & Niemi (1978), "A Multistage Game Representation of Sophisticated Voting for Binary Procedures", *J. Econ. Theory* 18 1-22.

[11] Moore (1992), "Implementation, Contracts, and Renegotiation in Environments with Complete Information", in *Advances in Economic Theory: Sixth World Congress, Vol. 1* (J-J Laffont ed.), New York: Cambridge University Press.

[12] Moore & Repullo (1988), "Subgame Perfect Implementation", *Econometrica* 56 1191-1220.

[13] Moulin (1979), "Dominance-Solvable Voting Schemes", *Econometrica* 47 1337-1351.

[14] Moulin (1986), "Choosing from a Tournament", *Soc. Choice Welfare* 3 271–291.

[15] Miller (1977), "Graph-Theoretical Approaches to the Theory of Voting", *Am. J. Polit. Sci.* 21 769–803.

[16] Palfrey & Srivastava (1989), "Nash Implementation Using Undominated Strategies", *Econometrica* 59 479–401.

[17] Srivastava & Trick (1996), "Sophisticated Voting Rules: The Case of Two Tournaments", *Soc. Choice Welfare* 13 275–289.

[18] Trick (2006), "Small Binary Voting Trees", Carnegie Mellon University *mimeo.*

[19] Trick (2009), "An Operations Research Look at Voting", Algorithmic Decision Theory Conference (Venice) *mimeo.*

[20] Vartiainen (2007a), "Subgame Perfect Implementation: A Full Characterization", *J. Econ. Theory* 133 111-126.

[21] Vartiainen (2007b), "Subgame Perfect Implementation of Voting Rules via Randomized Mechanisms", *Soc. Choice Welfare* 29 353-367.

Sean Horan
Département des sciences économiques
Université du Québec à Montréal (UQAM)
315 Ste-Catherine est, Montréal, QC, Canada
Email: `horan.sean@uqam.ca`

Strategic Behavior in a Decentralized Protocol for Allocating Indivisible Goods

Thomas Kalinowski, Nina Narodytska, Toby Walsh and Lirong Xia

Abstract

We study in detail a simple sequential procedure for allocating a set of indivisible goods to multiple agents. Agents take turns to pick items according to a policy. For example, in the alternating policy, agents simply alternate who picks the next item. A similar procedure has been used by Harvard Business School to allocate courses to students. We study here the impact of strategic behavior on the complete-information extensive-form game of such sequential allocation procedures. We show that computing the subgame-perfect Nash equilibrium is PSPACE-hard in general, but takes only linear time with two agents. Finally we compute the optimal policies for two agents in different settings, including when agents behave strategically and when agents can give away items.

1 Introduction

Suppose you are coaching a soccer team. To divide the players into two teams, you select the two best players as captains and then let them alternate at picking the remaining team members. Is this the best way to get an evenly matched game? Perhaps it would be better to reverse the order of their picks every round (so that the captain who picks first in the first round picks second in the second round)? This is an example of a problem in allocating indivisible goods. A number of real world problems involve allocating indivisible goods "fairly" between competing agents subject to possibly different preferences for these goods. For example, assigning courses to students at a business school is a problem of allocating indivisible goods. Students are competing for places on the popular courses, but have different preferences as to which courses to study. As a second example, the allocation of landing and take-off slots at an airport is a problem of allocating indivisible goods. Airlines are competing for popular landing and take-off times, but have different preferences as to precisely which slots they want. As a third and final example, sharing time slots on an expensive telescope is a problem of allocating indivisible goods. Astronomers are competing for observation time but have different preferences as to precisely which time slots are useful for their experiments.

Different properties might be demanded of a procedure for allocating indivisible goods. For example, we might look for allocations which are envy-free in the sense that every agent likes their allocation at least as much as the allocation to any other agent. However, envy-freeness by itself is not sufficient to ensure a "good" allocation. Not allocating any items is envy-free, and there are also many situations where no envy-free allocation exists. We might consider other criteria including efficiency of the allocation (e.g. Pareto optimality) and truthfulness of the mechanism (e.g. can agents profit by acting strategically?). There is, however, a tension between these properties. Svensson showed that the only strategy-proof, non-bossy[1] and neutral mechanism is a serial dictatorship in which agents take turns according to some order to pick their complete allocation of goods [5]. Unfortunately, a serial dictatorship can have a low efficiency in the utilitarian or egalitarian sense. In this

[1] A mechanism is non-bossy if when an agent submits different preferences and their allocation does not change then the overall allocation does not change.

paper, we focus on efficiency, and consider the impact on efficiency of such issues like the strategic behavior of the agents.

2 Existing methods

Several sequential procedures for allocating indivisible goods have been proposed in the literature. For example, the Harvard Business School has been using a mechanism called *Draft* to allocate courses to students [3]. The Draft mechanism generates a priority order over all students uniformly at random. Courses are then allocated to students in rounds. In odd rounds, each student is assigned to their favourite course that still has availability using the priority order. In even rounds, the mechanism uses the reverse priority order. The Draft mechanism is not strategy-proof. Indeed, students at Harvard have been observed to behave strategically [3]. Such strategic behavior can be harmful to the ex post social welfare. However, the expected (ex ante) social welfare is higher than that of a strategy-proof mechanism like serial dictatorship. To obviate the need for certain types of manipulation, Kominers, Ruberry and Ullman [4] proposed a mechanism in which proxies play strategically. They prove that with lexicographic preferences, this proxy mechanism is Pareto efficient.

As a second example, Bouveret and Lang [1] consider a simple decentralized procedure from [2] which resembles the Draft mechanism (but ignores the initial randomisation of the order of the students). The procedure is parameterised by a policy, the sequence in which agents take turns to pick items. This policy is fixed and assumed to be known to the agents in advance. For example, as in the Draft mechanism, with two agents and four items, the policy 1221 gives first and last pick to the first agent, and second and third pick to the second agent. This procedure has the advantage that the preferences of the agents do not need to be explicitly elicited. Bouveret and Lang assume agents have additive utilities given by a common scoring function (e.g. Borda, lexicographic or quasi-indifferent scores). They consider two extreme cases: full correlation in which preference orderings of the agents are identical, and full independence in which all preference orderings are equally probable. With full correlation, all policies give the same expected sum of utilities, and the sequential allocation procedure is strategy proof. With lexicographic scores, they show that the optimal strategy for an agent given a particular policy can be computed in polynomial time supposing other agents pick truthfully. The contribution of our paper is to study this sequential allocation procedure in more detail.

3 Preliminaries

Let $\mathcal{I} = \{c_1, \ldots, c_m\}$ denote a set of m indivisible goods, and $\mathcal{A} = \{A_1, \ldots, A_n\}$ denote a set of n agents. For any $j \leq n$, let $u_j : \mathcal{I} \rightarrow \mathcal{R}$ denote the utility function of agent A_j over $\mathcal{I}$. We assume $m \geq n$, and all agents have strict preferences. That is, for any $j \leq n$ and any pair of items $\{c, c'\}$, $u_j(c) \neq u_j(c')$. We suppose that an agent's utility function is *additive*. For any $j \leq n$ and any set of items $G \subseteq \mathcal{I}$, $u_j(G) = \sum_{c \in G} u_j(c)$. For any $j \leq n$, let O_j denote the ordinal preferences of agent j. That is, O_j is a total strict order over $\mathcal{I}$ and for any pair of items $\{c, c'\}$, $c \succ c'$ in O_j if and only if $u_j(c) > u_j(c')$. An agent has *Borda* utility, if for any $i \leq m$, the utility of the item ranked in i-th position in O_j is $m - i$. An agent has *lexicographic* utility, if for any $i \leq m$, the utility of the item ranked in i-th position in O_j is 2^{m-i}. An *allocation* is a function $f : \mathcal{I} \rightarrow \mathcal{A}$. For any agent $A \in \mathcal{A}$, $f^{-1}(A)$ denotes the set of items allocated to A. A sequential allocation is a mechanism parameterised by a *policy* P. This can be represented by an ordering over m elements taken from $\mathcal{A}$ (e.g. $P = [A_1 \succ A_2 \succ A_1]$). Agents take turns to pick items according to this ordering.

4 Optimal Policies

Bouveret and Lang considered which policies maximise the social welfare of the agents supposing the preference of agents are independent and every preference ordering is equally likely [1]. They considered a utilitarian principle in which social welfare is measured by the expected sum of the utilities of the agents (ExpSumUtil). They demonstrated that the simple alternating policy 121212... optimises the social welfare when utilities are Borda score (i.e. where the ith ranked of m items has a utility of $m - i$) and up to 12 items. Interestingly, there exist situations where the policy that maximises the sum of the utilities is not alternating. In fact, it need not even be balanced (that is, it might not assign an equal number of items to both agents).

Example 1. *Consider 8 items, a to h, 2 agents and utilities which are Borda scores. Suppose agent 1 has the preference order $a > \ldots > h$ whilst agent 2 has the order $a > h > b > c > d > e > f > g$. Then, supposing the agents pick items truthfully, the alternating policy* 12121212 *gives a social welfare of 22+16=38 but the optimal policy is* 22111111 *which gives a social welfare of 27+15=42. Note that the optimal policy does not Pareto dominate the alternating policy since, whilst the optimal policy increases the utility for agent 1, the utility for agent 2 decreases slightly.*

Of course, an alternating policy can still be the best policy in expectation even if there are individual situations like the above where it is not the best. Bouveret and Lang also considered an egalitarian principle in which social welfare is measured by the minimum of the expected utilities of the different agents (MinExpUtil). We consider two more other measures of egalitarian social welfare: the expected minimum utility of the different agents (ExpMinUtil) and the minimum utility of the different agents over all possible worlds (MinUtil). In the economics literature, MinExpUtil is called the ex-ante egalitarian utility, whilst ExpMinUtil is called the ex-post egalitarian utility.

To illustrate the difference between the three measures, consider the following two protocols. In the first, we toss a coin. If it lands on heads, we assign all m items to agent 1, otherwise we assign all items to agent 2. In the second protocol, we assign $m/2$ items at random to agent 1 and the rest to agent 2. The second protocol is more egalitarian than the first since one agent is sure to get no items in the first protocol whilst each agent is allocated $m/2$ items in the second protocol. This is reflected in the expected minimum of the two utilities (which is zero for the first protocol and half the total utility for the second protocol), and in the minimum utility (which is zero for the first protocol, and the sum of utilities of the least valuable $m/2$ items for the second protocol). However, the minimum of the expected utilities hides this difference as both protocols have a minimum expected utility that is half the total. We have the following proposition, whose proof is straightforward and is omitted.

Proposition 1. *For any policy and any distribution over utility functions:* MinUtil≤ExpMinUtil≤MinExpUtil

Note that, whilst the minimum utility (MinUtil) occurs in the full correlation case where agents utilities are identical [1], it can also occur when the utilities of the agents are different. For instance, suppose we are dividing just two items between two agents. Consider the protocol where the two agents declare which of the two items that they like most. If the two agents most prefer the same item, then we toss a coin to decide which agent gets this item, and assign the remaining, less preferred item to the other agent. On the other hand, if the two agents most prefers different items, we toss a coin and assign both items to an agent chosen at random. The minimum utility is now zero and occurs when the two agents most prefer different items. The full correlation case increases MinUtil to the smallest utility assigned to either object.

For the case of two agents, we computed the policies that maximise the three different egalitarian measures of social welfare using brute force search. Table 1 demonstrates that the optimal policies for maximising `ExpMinUtil` and `MinExpUtil` differ. We conjecture that the optimal `ExpMinUtil` policy has the form: $(12)^k2$ for $m = 2k+1$, $(12)^k(21)^k$ for $m = 4k$ and $(12)^k(21)^{k-1}$ for $m = 4k-2$. In addition, we conjecture that the optimal `ExpMinUtil` policy for an even number of items is also an optimal `MinUtil` policy.

m	MinExpUtil	ExpMinUtil	MinUtil
1	1	1	1
2	12	12	12
3	122	122	122
4	1221	1221	1221
5	*11222*	12122	12122, 12212, 12211
6	121221	121221	121221, 121221, 121222, 122121, 122112
7	*1122122*	1212122	*1212212, 1212212, 1221122, 1221211*
8	*12212112*	12122121	*11222122, 11222211,* . . . , 12212112, *12221111*

Table 1: Optimal policies that maximise the minimum of the two expected utilities (`MinExpUtil`), the expected minimum of the two utilities (`ExpMinUtil`) and the minimum utility (`MinUtil`). In each case, we allocate m items, assign utilities using *Borda* scoring, and assume full independence between the two agents. *Emphasis* is added to highlight when policies start to differ.

To return to our soccer example, suppose there are ten players to divide into two teams, utilities are Borda scores, and we adopt an egalitarian position to help ensure a balanced match. We might then select the two best players as team captains and, based on the optimality of the policy 12122121, have the first team captain pick first, third, sixth and eight, and the second team captain pick otherwise.

m	ExpMinUtil egalitarian	MinUtil egalitarian	ExpSumUtil utilitarian
1	1	1	1
2	12	12	12
3	122	122	121
4	1221	1222	1212
5	12122	12222	12121
6	122121	122222	121212
7	1221211	1222222	1212121
8	12212112	12222222	12121212

Table 2: Optimal policies that maximise the expected minimum of the utilities (ExpMinUtil), maximise the minimum utility (MinUtil) and maximise the expected sum of utilities (ExpSumUtil). In each case, we allocate m objects, assign utilities using *lexicographic* scoring, and assume full independence between the two agents.

As in [1], we also considered two other scoring models: lexicographic scoring (where an item at position k is scored 2^{-k}) and quasi-indifferent (where an item at position k is scored $a-k$ for $a \gg m$). We consider both an egalitarian model (the ExpMinUtil and MinUtil

m	ExpMinUtil egalitarian	MinUtil egalitarian	ExpSumUtil utilitarian
1	1	1	1
2	12	12	12
3	122	122	121
4	1221	1221	1212
5	11222	11222	12121
6	121221	121221, 122112, 122121	121212
7	1112222	1112222	1212121
8	12122121	11222211, 12122121, 12211221, 1221211	12121212

Table 3: Optimal policies that maximise the expected minimum of the utilities (ExpMinUtil), maximise the minimum utility (MinUtil) and maximise the expected sum of utilities (ExpSumUtil). In each case, we allocate m objects, assign utilities using quasi-indifferent scoring, and assume full independence between the two agents.

policies in which we maximise the expected or actual minimum utilities) and a utilitarian model (the ExpSumUtil policy in which we maximise the expected sum of the utilities). In Tables 2 and 3, we report the optimal policies for lexicographic and quasi-indifferent scoring.

We make some observations about these results. First, in both scoring models, a simple alternating policy is optimal under the utilitarian assumption. It seems likely that the expected sum of utilities is maximised for a wide variety of scoring functions by this policy. Second, for the quasi-indifferent scoring function, the same policy is optimal for ExpMinUtil and MinUtil. This was not the case for the lexicographic scoring model. For Borda scoring, the same policy was optimal for ExpMinUtil and MinUtil only for even n.

5 Strategic Behavior

Another desirable property of an allocation procedure is strategy-proofness. A sequential allocation procedure is strategy-proof if for any utility functions, the agents are best off choosing their top ranked item still available at every step. Unfortunately, the sequential allocation procedure is not strategy-proof in general. For instance, the first agent to pick an item might not pick their most preferred item if this is the item least preferred by the other agent. The first agent might strategically pick some other item as the second agent will not pick this first item unless there is no other choice. Bouveret and Lang [1] argue that the sequential allocation procedure is strategy-proof when agents have the same preference rankings. They also gave a polynomial time method for a single agent to compute a manipulation supposing all other agents act truthfully and utilities are lexicographic. Supposing all agents but the manipulator act truthfully is a strong assumption. If one agent is acting strategically, why not the others?

The sequential allocation procedure naturally lends itself to a game theoretic analysis in which all agents can act strategically. Assuming that the agents know the utility functions of other agents, we can model the sequential allocation procedure as a complete information *extensive-form game*. The subgame-perfect Nash equilibrium (SPNE) gives the (perhaps untruthful) strategy in which agents cannot improve their allocation by deviating unilaterally. The SPNE can be computed by *backward-induction* as follows. We start with the last agent A in the order P. For any allocation of items in the previous rounds, only one item remains, and A will get it. Then, we move to the second to the last agent A' in P.

For any allocation of items in previous round, A' can predict the final allocation for any item she picks. Therefore, she can pick an item that maximises her total utility in the final allocation. We then move on to the third to the last agent in P, etc. Since an agent can obtain the same total utility for picking different items, there might be multiple SPNE.

Example 2. *Suppose there are two agents and four items. Agent 1's ordinal preferences are $O_1 = c_1 \succ c_2 \succ c_3 \succ c_4$ and agent 2's ordinal preferences are $O_2 = c_2 \succ c_3 \succ c_4 \succ c_1$. Let $P = A_1 \succ A_2 \succ A_2 \succ A_1$. If all agents behave truthfully, then A_1 chooses c_1 in the first round, A_2 chooses c_2 and c_3 in the second and third rounds, respectively, and A_1 chooses c_4 in the last round. If the agents behave strategically, then A_1 can choose c_2 in the first round, and still get c_1 in the last round. The unique SPNE allocation in this game has A_1 getting $\{c_1, c_2\}$ and A_2 getting $\{c_3, c_4\}$.*

In the above example, even though there are multiple SPNE, the final allocation is unique regardless of the utility functions. We will see later that this is not a coincidence. When there are two agents, the *final* SPNE allocation is always unique (and indeed can be computed in linear time). The next example shows that with three or more agents, there can be multiple SPNE allocations.

Example 3. *Suppose there are four items and three agents with Borda utilities. The ordinal preferences of the agents are as follows. $A_1 : c_1 \succ c_2 \succ c_3 \succ c_4$, $A_2 : c_3 \succ c_4 \succ \ldots$, and $A_3 : c_1 \succ c_2 \succ \ldots$. Let $P = A_1 \succ A_2 \succ A_3 \succ A_1$. There are two SPNE allocations: (1) if A_1 picks c_1 in the first round, then in the SPNE A_1 gets $\{c_1, c_4\}$, A_2 gets c_3, and A_3 gets c_2; (2) if A_1 picks c_3 in the first round, then in the SPNE A_1 gets $\{c_2, c_3\}$, A_2 gets c_4, and A_3 gets c_1.*

5.1 Computing SPNE for Two Agents

With two agents and m items, computing the subgame-perfect Nash equilibrium by backward induction takes $\Omega(m!)$ time. This will be prohibitive when we have many items. The SPNE can, however, be computed in just $O(m)$ time by means of the following result. Let u_1, u_2 be the utility functions of the two agents, O_1, O_2 be their ordinal preferences, and P be the policy. We let $\text{Seq}(O_1, O_2, P)$ denote the truthful sequential allocation. We use $\text{SPNE}(u_1, u_2, P)$ to denote the subgame-perfect Nash equilibrium allocation. For any total strict order O, let $rev(O)$ denote the reversed order. Then, we can show that the SPNE allocation is unique, and can be computed from the truthful sequential allocation for the reversed preference orderings and policy.

Theorem 1. *When there are two agents, the SPNE allocation is unique. Moreover, $SPNE(u_1, u_2, P) = Seq(rev(O_2), rev(O_1), rev(P))$*

Proof: (Sketch) W.l.o.g. suppose agent 1 has the last pick in policy P (and thus the first pick in policy $rev(P)$). Then, agent 1 knows that the item that is ranked last in O_2 is "safe", as agent 2 has no incentive to pick it in earlier rounds. Therefore, agent 1 can safely pick this item in her last round, and leave opportunities in previous rounds in P to pick more popular items. The formal proof is much more involved and is proved by induction on the number of items m. ♣

Example 4. *Suppose there are two agents and four items. The agents' preferences and the policy are the same as in Example 2. We have $rev(P) = P$. In $Seq(rev(O_2), rev(O_1), rev(P))$, A_1 picks c_1 in the first round, A_2 picks c_3 and c_4 in the second round and third round respectively, and A_1 picks c_2 in the last round. This outcome is the same as the SPNE allocation in Example 2.*

5.2 Computing SPNE for more than Two Agents

When the number of agents n is comparable to the number of items m (more precisely, when $n = O(m)$), we prove that computing the SPNE is intractable. Consider the decision problem SubgamePerfect, where we are given the utility functions of n agents over m items, a particular agent A, a policy P, and a threshold T, and we are asked whether the utility of A is larger than T in any SPNE. Computing the SPNE for a finite multi-player extensive game with perfect information is naturally in PSPACE [6]. Our contribution here is to show that this particular game is complete for this complexity class.

Theorem 2. SubgamePerfect *is PSPACE-complete for Borda scoring of utilities.*

Proof: Backward induction shows that it is in PSPACE. To show hardness, we give a reduction from QSAT, which is a standard PSPACE-complete problem. In a QSAT instance, We are given a quantified formula $\exists x_1 \forall x_2 \exists x_3 \cdots \forall x_q \,.\, \varphi$ where q is even and we are ask whether the formula is true. Let $\varphi = C^1 \wedge \cdots \wedge C^t$, where C^j is a 3-clause, $l_j^1 \vee l_j^2 \vee l_j^3$. We construct a SubgamePerfect instance where there is a unique SPNE with a utility to the first player larger than a threshold if and only if the formula is true.

In the SubgamePerfect instance, there are q agents who represent the binary variables. Each of these agents choosing one out of two items represents a valuation of the variable. The agents that correspond to $\exists$ quantifiers (that is, agents 1, 3,…, $q-1$) obtain higher utility if φ is true under the current valuation, and the agents that correspond to $\forall$ quantifiers (that is, agents 2, 4,…, q) obtain higher utility if φ is false under the current valuation. There are also some other agents that are used to encode the QSAT instance, which we will specify later.

Let a be an item, and k, p be natural numbers. We define an ordering $O_p^k(a)$ that will be used as part of the policy P as follows. It introduces $2k+1$ new agents $A_p^1, \ldots, A_p^{2k+1}$ and $5k+1$ new items $\{a_p, b_p^1, \ldots, b_p^k, c_p^1, \ldots, c_p^k, d_p^1, \ldots, d_p^k, e_p^1, \ldots, e_p^k, f_p^1, \ldots, f_p^k\}$. The preferences of the new agents are as follows:

Agent	Preferences
A_p^1	$b_p^1 \succ c_p^1 \succ d_p^1 \succ e_p^1 \succ$ Others
$\vdots$	$\vdots$
A_p^k	$b_p^k \succ c_p^k \succ d_p^k \succ e_p^k \succ$ Others
A_p^{k+1}	$c_p^1 \succ f_p^1 \succ$ Others
$\vdots$	$\vdots$
A_p^{2k}	$c_p^k \succ f_p^k \succ$ Others
A_p^{2k+1}	$a \succ b_p^k \succ \cdots \succ b_p^1 \succ a_p \succ$ Others

Let the order over agents be $A_p^1 \succ \cdots \succ A_p^{2k+1} \succ A_p^1 \succ \cdots \succ A_p^2$. In $O_q^k(a)$, a is the item that we want to "duplicate", k is the number of duplicates, and q is merely an index. We can prove by induction that if a has not been chosen (in previous rounds), then after agents have chosen items according to $O_q^k(a)$, $\{f_p^1, \ldots, f_p^k\}$ will be chosen and $\{d_p^1, \ldots, d_p^k\}$ will not be chosen; if a has been chosen, then $\{d_p^1, \ldots, d_p^k\}$ will be chosen rather than $\{f_p^1, \ldots, f_p^k\}$.

We now specify the sequential allocation instance by using the orderings $O_p^k(a)$. All agents introduced in $O_p^k(a)$ will not appear in other places in the policy P. For each $i \le q$, there are two items 0_i and 1_i that represent the two values of x_i, an agent A_i corresponding to the valuation and another agent B_i that is used to make sure that A_i chooses 0_i or 1_i in the $(q+2i-1)$th round. For each $i \le q$, D_i is an agent whose preferences are $d_i \succ$ Others, where d_i is a new item that creates a "gap" between items available to agent A_i. The first $(2t+4)q$ agents in P are the following: $D_1 \succ \cdots \succ D_q \succ A_1 \succ \cdots \succ A_q \succ O_1^t(0_1) \succ \cdots \succ$

$O_q^t(0_q) \succ B_1 \succ \cdots \succ B_q$. The preferences of B_i are $0_i \succ 1_i \succ$ Others. The preferences of A_i will be defined after we have defined all items and have specified P. For notational convenience, for each $i \leq q$ and each $j \leq t$ we rename d_i^j to be 0_i^j, and rename f_i^j to be 1_i^j.

For each clause C^i, we have an agent denoted by C_i. Suppose v_{j_1}, v_{j_2}, and v_{j_3} correspond to the 3 valuations that satisfy C_i, then we let the preferences of C_i be $v_{j_1}^i \succ v_{j_2}^i \succ v_{j_3}^i \succ g \succ g_i' \succ$ Others, where g and g_i' are new items. g is used to detect whether a clause is not satisfied. For example, suppose $C^i = x_1 \vee \neg x_2 \vee x_3$, then the preferences of C_i are $1_1^i \succ 0_2^i \succ 1_3^i \succ g \succ g_i' \succ$ Others. The remaining agents in the P are: $C_1 \succ \cdots \succ C_t \succ O_{q+1}^q(g) \succ A_1 \succ \cdots \succ A_q$.

The agents and new items introduced in $O_{q+1}^q(g)$ impose "feedback" on A_1 through A_q, such that if g is allocated before $O_{q+1}^q(g)$ (which means that the formula is not satisfied under the valuation encoded in the first q rounds), then some items that are more valuable to the agents that correspond to the $\forall$ quantifiers are made available; if g is not allocated before $O_{q+1}^q(g)$, then some items that are more valuable to the agents that correspond to the $\exists$ quantifiers are made available. Finally, for each $i \leq q$, we define the ordinal preferences of A_i as follows. If i is odd, then A_i's preferences are $0_i \succ 1_i \succ d_{q+1}^i \succ d_i \succ f_{q+1}^i \succ \ldots$. If i is even, then A_i's preferences are $0_i \succ 1_i \succ f_{q+1}^i \succ d_i \succ d_{q+1}^i \succ \ldots$.

To summarise, in the sequential allocation instance, there are $3q + t + (2t+1)q + 2q + 1$ agents and $m = 3q + (5t+1)q + 1 + t + 5q + 1$ items, which are polynomial in the size of the formula ($\Omega(t+q)$). Table 4 summaries the items introduced in the reduction. Final, the

for	items	Introduced in
$i \leq q$	d_i	D_i
$i \leq q$	$0_i, 1_i$	A_i
$i \leq q$, $j \leq t$	a_i b_i^j c_i^j d_i^j (a.k.a. 0_i^j) e_i^j f_i^j (a.k.a. 1_i^j)	$O_i^j(0_i)$
	g	C_1
$j \leq t$	g_t'	C_j
$j \leq q$	a_{q+1}, b_{q+1}^j, c_{q+1}^j, d_{q+1}^j, e_{q+1}^j, f_{q+1}^j	$O_{q+1}^q(g)$

Table 4: Items introduced in the reduction.

policy P ordering over agents is the following.

$$
\begin{aligned}
&D_1 \succ \cdots \succ D_q \succ A_1 \succ \cdots \succ A_q \succ O_1^t(0_1) \succ \cdots \succ O_q^t(0_q) \\
&\succ B_1 \succ \cdots \succ B_q \succ C_1 \succ \cdots \succ C_t \succ O_{q+1}^q(g) \\
&\succ A_1 \succ \cdots \succ A_q
\end{aligned}
$$

If we must allocate all items then we can add some dummy agents to the end of the ordering.

We note that if an agent only appears once in the ordering, then it is her strictly dominant strategy to pick her most preferred available item. In any SPNE, in the first q rounds $d_1, \ldots, d_q$ will be chosen. In the next q rounds, agent i must choose either 0_i or 1_i, otherwise 0_i will be chosen by agent A_i^{2t+1} introduced in $O_i^t(0_i)$ and 1_i will be chosen by B_i. Hence, the choices of agents A_i correspond to valuations of the variables, and these valuations are duplicated by $O_i^t(0_i)$ that will be used to satisfy clauses. (We note that if A_i chooses 0_i, then after $O_i^t(0_i)$, $\{0_i^1, \ldots, 0_i^t\}$ are still available, but $\{1_i^1, \ldots, 1_i^t\}$ are not available; and vice versa.) Then, a clause C^i is satisfied if and only if at least one of the top 3 items of agent C_i is available (otherwise C_i chooses g). Hence, after agent C_t, g is available if and only if all clauses are satisfied. Finally, if g is available after agent C_t, then the agents that correspond to the $\exists$ quantifiers can choose d_{q+1}'s to increase their total utility by $m-3$, but the agents that correspond to the $\forall$ quantifiers can only choose d_{q+1}'s to increase their utility by $m-5$; and vice versa. Hence, the agents that correspond to $\exists$ quantifiers will choose valuations to make F true, while the agents that correspond to $\forall$ quantifiers will choose valuations to make F false. It can be verified that there is a unique SPNE allocation, where agent A_1's utility is at least $2m-5$ (that is, she gets one of $\{0_1, 1_1\}$ and d_{q+1}^1) if and only if the formula F is true. ♣

6 Optimal Policies for Strategic Behavior

Suppose agents act strategically instead of truthfully. For example, suppose they pick items according to the subgame-perfect Nash equilibrium. The policies which maximise social welfare can now change. For a reversal symmetric scoring function like Borda, and a reversal symmetric policy like the simple alternating policy, the situations where strategic behavior decreases social welfare are exactly balanced by the symmetric situations where it increases social welfare. As a result, we did not observe any difference in the policies that optimises social welfare for Borda scoring when agents behave strategically instead of truthfully. For example, brute force calculation with up to 8 items show that the expected sum of the utilities of the agents supposing Borda scoring is maximised by the same simple alternating policy whether agents pick either truthfully or strategically.

Strategic behavior can sometimes increase the social welfare of the agents. In other cases, it can decrease the social welfare of the agents or leave it unchanged. In fact, given the reversal symmetry of the optimal policy, and of the subgame perfect equilibrium, Borda scoring and the utilitarian criterion, we can prove that the cases when the utilitarian social welfare increases are exactly matched by cases where it decreases. With an egalitarian criterion, strategic behavior can improve social welfare slightly more often than it can decrease it. Averaged over all possible preference profiles, brute force calculations suggest that the expected sum of the utilities barely changes, whilst the expected minimum increases by less than 1%.

For scoring functions that are not symmetric, the optimal policy can change. For example, with lexicographic scores, the optimal policy for strategic behavior is different from that for truthful behavior. Table 5 summarises results based on brute force calculation. When maximising the expected minimum utility, the optimal policies for agents playing strategically are optimal policies for agents playing truthfully for 6 or fewer items. However, the optimal policy for strategic play with 7 items is 1221122 but for truthful play is 1221211. Similarly, for 8 items, the optimal policy for strategic play is 12212211 but for truthful play is 12212112. When maximising the minimum utility, the optimal policies for strategic play are optimal policies for truthful play. When maximising the expected sum of utilities and 4 or more items, the optimal policies for strategic play are not optimal alternating policies for truthful play.

We conjecture that the optimal `ExpMinUtil` policy supposing strategic behavior has the alternating form: $(1221)^k21$ for $m = 4k+2$, $(1221)^k122$ for $m = 4k+3$ and $(1221)^k2211$ for $m = 4k+4$. We also conjecture that the optimal `ExpSumUtil` policy supposing strategic behavior has the alternating form: $(12)^k122$ for $m = 2k+3$, $1(2211)^k2$ for $m = 4k+2$, and $1(2211)^k221$ for $m = 4k+4$. Strategic play also carries a small cost. Averaged over all possible preference profiles, the utility decreases by 5% or less for both the expected sum

n	`ExpMinUtil` egalitarian	`MinUtil` egalitarian	`ExpSumUtil` utilitarian
1	1	1	1
2	12	12	12
3	122	122	121
4	1221	1222	1212, *1221*
5	12122	12222	*12122*
6	122121	122222	*122112*
7	*1221122*	1222222	*1212122*
8	*12212211*	12222222	*12211221*

Table 5: Optimal policies when we assign utilities using lexicographic scoring, and assume agents play strategically by computing the subgame-perfect Nash equilibrium. *Emphasis* is added to highlight when policies differ from the optimal truthful policies.

and minimum of utilities.

As in [1], we also considered quasi-indifferent scoring. With quasi-indifferent scoring, an item at position k in an agent's ordering is given score $a - k$ where $a \gg m$. In Table 6, we give the optimal policies for agents playing strategically when agents are quasi-indifferent between items. The optimal policy for agents playing strategically is also the optimal policy for agents playing truthfully except $m = 6$ and the egalitarian criterion of maximising the expected minimum utility. When agents play strategically, the optimal policy in this case is 122121. However, when agents play truthfully, the optimal policy in this case is 121221.

m	`ExpMinUtil` egalitarian	`MinUtil` egalitarian	`ExpSumUtil` utilitarian
1	1	1	1
2	12	12	12
3	122	122	121
4	1221	1221	1212
5	11222	11222	12121
6	122121	121221, 122112, 122121	121212
7	1112222	1112222	1212121
8	12122121	12211221, 12212112, 12122121, 11222211	12121212

Table 6: Optimal policies when we assign utilities using a quasi-indifferent scoring function, and assume agents play strategically by computing the subgame perfect equilibrium.

7 Disposal of Items

One inefficiency of the policies considered so far is that one agent may use one of their early choices to select an item that the other agent would happily give away. There is an inherent asymmetry in agents declaring items that they like most but not the items that they like least. To address this issue, we suppose agents can select the item that they least like to give to the other agent. For instance, the policy $1\bar{1}21$ describes a protocol in which the first agent starts by picking their most preferred item, then picks their least preferred item to give to the second agent, the second agent then picks the most preferred of the two items that remain, and the first agent then gets the last remaining item. $\bar{1}$ means that agent

1 gives the item remaining that she likes least to agent 2. Such disposal of items can be extended to more than 2 agents but requires a protocol for which agent takes the disposed item.

m	ExpMinUtil egalitarian	ExpSumUtil utilitarian
1	1	1
2	12	12
3	122	121, $\bar{1}21$
4	1221, $1\bar{1}21$, $1\bar{2}22$, $\bar{1}211$	$1\bar{1}21$
5	12122, $\bar{1}\bar{1}\bar{2}12$	$12\bar{2}12$, $\bar{1}2\bar{2}12$
6	$1\bar{2}\bar{1}\bar{1}21$, $\bar{1}21121$	$1\bar{1}2121$, $1\bar{1}\bar{2}\bar{1}21$
7	$12\bar{1}\bar{1}\bar{2}12$	$1212\bar{2}12$, $1\bar{2}\bar{1}2\bar{2}12$, $\bar{1}212\bar{2}12$, $\bar{1}\bar{2}\bar{1}2\bar{2}12$
8	$1\bar{2}\bar{1}\bar{1}\bar{2}\bar{1}21$, $\bar{1}2112121$	$1\bar{1}212121$, $1\bar{1}21\bar{2}\bar{1}21$, $1\bar{1}\bar{2}\bar{1}2121$, $1\bar{1}\bar{2}\bar{1}\bar{2}\bar{1}21$

Table 7: Optimal policy for dividing m items with utility measured using Borda scoring assuming egalitarianism or utilitarianism and full independence between the two agents. Note that when computing the optimal policy, we consider all possible policies including those in which agents only pick items, and those in which agents only give items away.

In Table 7, we give the optimal policies assuming strategic behavior, and Borda scoring of utilities when agents can dispose of items as well as pick them. We again put policies into a canonical form in which agent 1 makes the first move. There is a symmetric policy in which we swap agent 1 with agent 2 throughout. We also ignore policies which result in the same division of items. For instance, a policy containing the moves $\bar{1}1$ is equivalent one containing $1\bar{1}$. Our canonical form has agents picking items before giving give them away. For example, a policy that ends with the moves $\bar{2}1$ gives the last two items to the first agent so is equivalent to one that ends with the moves 11. Our canonical form describes a policy by the lexicographically least equivalent policy supposing that 1 and 2 are ordered before $\bar{1}$ and $\bar{2}$.

We make some observations about the results. First, we can often increase social welfare by having agents declare items that they dislike. There are a few optimal policies in which agents only pick items that they like (e.g. for $m = 5$, one of the optimal egalitarian policies is 12122). However, in most cases, the optimal policy has agents declaring both items that they like and dislike. Second, when dividing 4 items between two agents, there is a policy, $1\bar{1}21$ that is optimal for both the egalitarian and utilitarian measures of social welfare. Third, unlike protocols in which agents pick just items that they like, there are often several different protocols which maximise social welfare.

8 Conclusions

We have studied a simple sequential allocation procedure where agents get to choose items according to a policy, and agents have simple additive utilities over items given by Borda, lexicographic or quasi-indifferent scores. We have computed optimal policies assuming both truthful and strategic behavior of the agents for both egalitarian and utilitarian measure of social welfare. We have also proved that with two agents, the subgame perfect Nash equilibrium is polynomial to compute by simply reversing the agents' preferences and the policy. On the other hand, with more than two agents, we proved that computing the subgame perfect Nash equilibrium is PSPACE-hard. There are many directions for future work. One direction would be to prove the conjectures about the optimal policies for maximising social welfare assuming truthful or strategic behavior and Borda or lexicographic scoring. Another direction would be to determine if we can compute the subgame-perfect Nash equilibrium in polynomial time for a fixed number agents k where $k > 2$. More generally, when we want to allocate multiple indivisible goods, how can we design simple decentralized mechanisms that balance efficiency and strategy-proofness?

References

[1] S. Bouveret and J. Lang. A general elicitation-free protocol for allocating indivisible goods. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 73–78. IJCAI/AAAI, 2011.

[2] S.J. Brams and A.D. Taylor. *Fair Division: From cake-cutting to dispute resolution.* Cambridge University Press, Cambridge, 1996.

[3] E. Budish and E. Cantillon. The Multi-Unit Assignment Problem: Theory and Evidence from Course Allocation at Harvard. *American Economic Review*, page (forthcoming), 2012.

[4] S.D. Kominers, M. Ruberry, and J. Ullman. Course allocation by proxy auction. In A. Saberi, editor, *6th International Workshop on Internet and Network Economics (WINE 2010)*, volume 6484 of *Lecture Notes in Computer Science*, pages 551–558. Springer, 2010.

[5] L.-G. Svensson. Strategy-proof allocation of indivisible goods. *Social Choice and Welfare*, 16(4):557–567, 1999.

[6] C. lvarez, J. Gabarr, and M. Serna. Polynomial space suffices for deciding Nash equilibria properties for extensive games with large trees,. In X. Deng and D.-Z. Du, editors, *Algorithms and Computation*, volume 3827 of *Lecture Notes in Computer Science*, pages 634–643. 2005.

Thomas Kalinowski
Universität Rostock
Rostock, Germany
Email: thomas.kalinowski@uni-rostock.de

Nina Nardodytska and Toby Walsh
NICTA and UNSW
Sydney, Australia
Email: {nina.narodytska,toby.walsh}@nicta.com.au

Lirong Xia
Harvard University
Cambridge, USA
Email: xialirong@gmail.com

Aggregating Conditionally Lexicographic Preferences on Multi-Issue Domains

Jérôme Lang, Jérôme Mengin, and Lirong Xia

Abstract

One approach to voting on several interrelated issues consists in using a language for compact preference representation, from which the voters' preferences are elicited and aggregated. A language usually comes with a domain restriction. We consider a well-known restriction, namely, *conditionally lexicographic preferences*, where both the relative importance between issues and the preference between values of an issue may depend on the values taken by more important issues. The naturally associated language consists in describing conditional importance and conditional preference by trees together with conditional preference tables. In this paper, we study the aggregation of conditionally lexicographic preferences, for several voting rules and several restriction of the framework. We characterizes computational complexity for some popular cases, and show that in many of them, computing the winner reduces in a very natural way to a MAXSAT problem.

1 Introduction

There are many situations where a group of agents have to make a common decision about a set of possibly interrelated issues, variables, or attributes. For example, this is the situation in the following three domains:

• *Multiple referenda*: there is a set of binary issues (such as building a sport centre, building a cultural centre etc.); on each of them, the group has to make a yes/no decision.

• *Committee elections*: there is a set of positions to be filled (such as a president, a vice-president, a secretary).

• *Group product configuration*: the group has to agree on a complex object consisting of several components.

Voting on several interrelated issues has been proven to be a challenging problem from both a social choice viewpoint and a computational viewpoint. If the agents vote separately on each issue, then paradoxes generally arise [6, 13]; this rules out this 'decompositional' way of proceeding, except in the restricted case when voters have separable preferences. A second way consists in using a sequential voting protocol: variables are considered one after another, in a predefined order, and the voters know the assignment to the earlier variables before expressing their preferences on later ones (see, e.g., [14, 15, 2]). This method, however, works (reasonably) well only if we can guarantee that there exists a common order over issues such that every agent can express her preferences unambiguously on the values of each issue at the time he is asked to report them. A third class of methods consists in using a language for compact preference representation, in which the voters' preferences are stored and from which they are aggregated. If the language is expressive enough to allow for expressing any possible preference relation, then the paradoxes are avoided, but at a very high cost, both in elicitation and computation. Therefore, when organizing preference aggregation in multiple interrelated issue, there will always be a choice to be made between (a) being prone to severe paradoxes, (b) imposing a domain restriction or (c) requiring a heavy communication and computation burden.

In this paper, we explore a way along the third class of methods. When eliciting, learning, and reasoning with preferences on combinatorial domains, a domain restriction often considered consists in assuming that preferences are lexicographic. Schmitt et al. [17] address the learning of lexicographic preferences, after recalling that the psychology literature shows evidence that lexicographic preferences are often an accurate model for human decisions [10]. Learning such preferences is considered further in [8, 18], and then in [3] who learn more generally *conditionally lexicographic preferences*, where the importance order on issues as well as the local preferences over values of issues can be conditional on the values of more important issues. The aggregation of lexicographic preferences over combinatorial domains has received very little attention (the only exception we know of is [1]). Yet it appears to be – at least in some contexts – a reasonable way of coping with multiple elections. It does imply a domain restriction, and arguably an important one; but, as explained above, domain restrictions seem to be the only way of escaping both strong paradoxes and a huge communication cost, and conditionally lexicographic preference models are not so restrictive, especially compared to the most common domain restriction, namely separability.

The generic problem of aggregating conditionally lexicographic preferences can be stated as follows. The set of alternatives is a combinatorial domain $\mathcal{X}$ composed of a finite set of binary issues.[1] We have a set of voters, each providing a conditionally lexicographic preference over $\mathcal{X}$ under the compact and natural form of a lexicographic preference tree (LP-tree for short) [3], which we will define soon; therefore, a (compactly represented) profile P consists of a collection of LP-trees. Since each LP-tree $\mathcal{L}$ is the compact representation of one linear order $\succ_{\mathcal{L}}$ over $\mathcal{X}$, there is a one-to-one correspondence between P and the (extensively represented) profile P^* consisting of a collection of linear orders over $\mathcal{X}$. Finally, for a given voting rule r, we ask whether there is a simple way to compute the winner, namely $r(P^*)$, where 'simple' means that the winner should be computed directly (and efficiently) from P and in any case we must avoid to produce P^* *in extenso*, which would require exponential space. For many cases where winner determination is computationally hard, we show that these problems can be efficiently converted to MAXSAT problems and thus be solved by sat solvers.

The rest of the paper is organized as follows. Conditionally lexicographic preferences and their compact representation by LP-trees are defined and discussed in Section 2. In Section 3 we state the problem considered in this paper, namely the aggregation of conditionally lexicographic preferences by voting rules. As we will see, some voting rules are better than others in this respect. In the paper we focus on three families of rules. First, in Section 4, k-approval rules: we show that for many values of k, we can give a quite satisfactory answer to our question above, even for our most general models. Note that by 'satisfactory' we do not necessarily mean "computable in polynomial time": for instance, when deciding whether a given alternative is a winner is NP-complete but can be easily translated into a compact maximum (weighted) satisfiability problem, for which efficient algorithms exist, we still consider the answer as (more or less) positive. In Section 5 we then focus on the Borda rule, and show that the answer to our question is satisfactory for some of the simplest LP-tree models, but less so for some general models. We also provide a natural family of scoring rules for which the answer is positive in all cases. Then in Section 6 we consider the existence of a Condorcet winner, and show that for Condorcet-consistent rules, and in particular Copeland and maximin, the answer tends to be negative. Finally, Section 7 is devoted to the specific case of LP-trees with fixed local preferences. Due to the space constraint, most proofs are omitted.

2 Conditionally Lexicographic Preferences and LP-Trees

Let $\mathcal{I} = \{X_1, \ldots, X_p\}$ ($p \geq 2$) be a set of *issues*, where each issue X_i takes a value in a binary *local domain* $D_i = \{0_i, 1_i\}$. The set of alternatives is $\mathcal{X} = D_1 \times \cdots \times D_p$, that is, an alternative is uniquely identified by its values on all issues. Alternatives are denoted by $\vec{d}, \vec{e}$ etc. For any $Y \subseteq \mathcal{I}$ we denote $D_Y = \prod_{X_i \in Y} D_i$. Let $L(\mathcal{X})$ denote the set of all linear orders over $\mathcal{X}$.

Lexicographic comparisons order pairs of outcomes $(\vec{d}, \vec{e})$ by looking at the attributes in sequence, according to their importance, until we reach an attribute X such that the value of X in

[1]The assumption that variables are binary is made for the sake of simplicity due to the space constraint. Most of our results would easily extend to the non-binary case.

$\vec{d}$ is different from the value of X in $\vec{e}$; $\vec{d}$ and $\vec{e}$ are then ordered according to the *local preference* relation over the values of X. For such lexicographic preference relations we need both an *importance* relation, between attributes, and *local preference* relations over the domains of the attributes. Both the importance between attributes and the local preferences may be conditioned by the values of more important attributes. Such lexicographic preference relations can be compactly represented by *Lexicographic Preference trees (LP-trees)* [3], described in the next section.

2.1 Lexicographic Preference Trees

An LP-tree $\mathcal{L}$ is composed of two parts: (1) a tree $\mathcal{T}$ where each node t is labeled by an issue, denoted by $\mathsf{Iss}(t)$, such that each issue appears once and only once on each branch; each non-leaf node either has two outgoing edges, labeled by 0 and 1 respectively, or one outgoing edge, labeled by $\{0, 1\}$. (2) A *conditional preference table* $\mathsf{CPT}(t)$ for each node t, which is defined as follows. Let $\mathsf{Anc}(t)$ denote the set of issues labeling the ancestors of t. Let $\mathsf{Inst}(t)$ (respectively, $\mathsf{NonInst}(t)$) denote the set of issues in $\mathsf{Anc}(t)$ that have two (respectively, one) outgoing edge(s). There is a set $\mathsf{Par}(t) \subseteq \mathsf{NonInst}(t)$ such that $\mathsf{CPT}(t)$ is composed of the agent's local preferences over $D_{\mathsf{Iss}(t)}$ for all valuations of $\mathsf{Par}(t)$. That is, suppose $\mathsf{Iss}(t) = X_i$, then for every valuation $\vec{u}$ of $\mathsf{Par}(t)$, there is an entry in the CPT which is either $\vec{u} : 0_i \succ 1_i$ or $\vec{u} : 1_i \succ 0_i$. For any alternative $\vec{d} \in \mathcal{X}$, we let the *importance order* of $\vec{d}$ in $\mathcal{L}$, denoted by $\text{IO}(\mathcal{L}, \vec{d})$, to be the order over $\mathcal{I}$ that gives $\vec{d}$ in $\mathcal{T}$. We use $\rhd$ to denote an importance order to distinguish it from agents' preferences $\succ$ (over $\mathcal{X}$). If in $\mathcal{T}$, each vertex has no more than one child, then all alternatives have the same importance order $\rhd$, and we say that $\rhd$ is the importance order of $\mathcal{L}$.

An LP-tree $\mathcal{L}$ represents a linear order $\succ_{\mathcal{L}}$ over $\mathcal{X}$ as follows. Let $\vec{d}$ and $\vec{e}$ be two different alternatives. We start at the root node t_{root} and trace down the tree according to the values of $\vec{d}$, until we find the first node t^* such that $\vec{d}$ and $\vec{e}$ differ on $\mathsf{Iss}(t^*)$. That is, w.l.o.g. letting $\mathsf{Iss}(t_{\text{root}}) = X_1$, if $d_1 \neq e_1$, then we let $t^* = t_{\text{root}}$; otherwise, we follow the edge d_1 to examine the next node, etc. Once t^* is found, we let $U = \mathsf{Par}(t^*)$ and let d_U denote the sub-vector of $\vec{d}$ whose components correspond to the nodes in U. In $\mathsf{CPT}(t^*)$, if $d_U : d_{t^*} \succ e_{t^*}$, then $\vec{d} \succ_{\mathcal{L}} \vec{e}$. We use $\mathcal{L}$ and $\succ_{\mathcal{L}}$ interchangeably.

Example 1 *Suppose there are three issues. An LP-tree $\mathcal{L}$ is illustrated in Figure 1. Let t be the node at the end of the bottom branch. We have* $\mathsf{Iss}(t) = X_2$, $\mathsf{Anc}(t) = \{X_1, X_3\}$, $\mathsf{Inst}(t) = \{X_1\}$, $\mathsf{NonInst}(t) = \{X_3\}$, *and* $\mathsf{Par}(t) = \{X_3\}$. *The linear order represented by the LP-tree is* $[001 \succ 000 \succ 011 \succ 010 \succ 111 \succ 101 \succ 100 \succ 110]$, *where* 000 *is the abbreviation for* $0_1 0_2 0_3$, *etc.* $\text{IO}(\mathcal{L}, 000) = [X_1 \rhd X_2 \rhd X_3]$ *and* $\text{IO}(\mathcal{L}, 111) = [X_1 \rhd X_3 \rhd X_2]$.

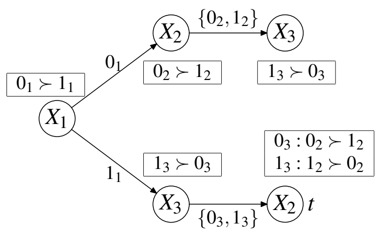


Figure 1: An LP-tree $\mathcal{L}$.

2.2 Classes of Lexicographic Preference Trees

The definition for LP-trees above is for the most general case. [3] also defined some interesting sub-classes of LP-trees by imposing a restriction on the local preference relations and/or on the conditional importance relation.

The local preference relations can be *conditional* (general case, as defined above), but can also be *unconditional* (the preference relation on the value of any issue is independent from the value of all other issues). The most restrictive case is *fixed*, which means that not only are the preferences unconditional, but that they are common to all voters. Formally, UP is the class of LP-trees with *unconditional local preferences*: for every issue X_i there exists a preference relation $\succ_i$ ($1_i \succ_i 0_i$ or $0_i \succ_i 1_i$) and for every node t with $X_i = \mathsf{Iss}(t)$, $\mathsf{Par}(t) = \emptyset$, and $\mathsf{CPT}(t) = \{\succ_i\}$. And FP is the class of LP-trees with *fixed local preferences* (FP): without loss of generality, for every node t (with $\mathsf{Iss}(t) = X_i$), $\mathsf{CPT}(t) = \{1_i \succ 0_i\}$.

Likewise, the importance relation over issues can be *conditional* (general case), or *unconditional*, of *fixed* when it is common to all voters: (UI) is the set of all linear LP-trees, i.e., every node has no more than one child. And (FI) is the set of all linear LP-trees with the (unconditional) importance order over issues $[X_1 \rhd \ldots \rhd X_p]$.

We can now combine a restriction on local preferences and a restriction on the importance relation. We thus obtain nine classes of LP-trees, namely, FI-FP, UI-FP, CI-FP, FI-UP, UI-UP, CI-UP, FI-CP, UI-CP, and CI-CP. For instance, UI-CP is defined as the class of all LP-trees with unconditional importance relation and conditional preferences. Note that the FI-FP class is trivial, as it contains a unique LP-tree.

Recall that a LP-tree is composed of a tree and a collection of conditional preference tables. The latter is reminiscent of CP-nets [4]. In fact, it can be viewed as some kind of generalized CP-net whose dependency relations between variables (induced from the importance relation) may be conditional on the values of their parent variables. However, in the case of an unconditional importance relation (UI), then the collection of CP-tables *is* a CP-net, and the LP-tree is a TCP-net [5]. In the general case however, a conditionally lexicographic preferences cannot be represented by a TCP-net.

3 Aggregating LP-trees by Voting Rules

We now consider n voters. A *(voting) profile* P over a set of alternatives $\mathcal{X}$ is a collection of n linear orders on $\mathcal{X}$. A *voting rule* r maps every profile P to a nonempty subset of $\mathcal{X}$: $r(P)$ is the set of *co-winners* for r and P.

A *scoring function* S is a mapping $L(\mathcal{X})^n \times \mathcal{X} \to \mathbb{R}$. Often, a voting rule r is defined so that $r(P)$ is the set of alternatives maximizing some scoring function S_r. In particular, *positional scoring rules* are defined via a *scoring vector* $\vec{v} = (v(1), \ldots, v(m))$, where m is the number of alternatives (here, $m = 2^p$): for any vote $V \in L(\mathcal{X})$ and any $c \in \mathcal{X}$, let $S_{\vec{v}}(V, c) = v(rank_V(c))$, where $rank_V(c)$ is the rank of c in V; then for any profile $P = (V_1, \ldots, V_n)$, let $S_{\vec{v}}(P, c) = \sum_{j=1}^{n} S_{\vec{v}}(V_j, c)$. The winner is the alternative maximizing $S_{\vec{v}}(P, \cdot)$. In particular, the k-approval rule App_k (with $k \le m$), is defined by the scoring vector $v(1) = \cdots = v(k) = 1$ and $v(k+1) = \cdots = v(m) = 0$, the scoring function being denoted by S^k_{App}; and the *Borda* rule is defined by the scoring vector $(m-1, m-2, \ldots, 0)$, the scoring function being denoted by S_{Borda}.

An alternative α is the *Condorcet winner* for a profile P if for any $\beta \neq \alpha$, a (strict) majority of votes in P prefers α to β. A voting rule is *Condorcet-consistent* if it elects the Condorcet winner whenever one exists. Two prominent Condorcet-consistent rules are *Copeland* and *maximin*. The Copeland winners are the alternatives α that maximize the Copeland score $C(\alpha)$, defined as the number of alternatives β such that a majority of votes in P prefers α to β. The maximin winners are the alternatives α that maximize the maximin score $S_{\mathrm{MM}}(\alpha)$, defined as $S_{\mathrm{MM}}(P, \alpha) = \max\{N_P(\beta, \alpha) : \beta \in \mathcal{X}, \beta \neq \alpha\}$, where $N_P(\beta, \alpha)$ denotes the number of votes in P that rank α ahead of β.

3.1 Voting Restricted to Conditionally Lexicographic Preferences

The key problem addressed in this paper is the following. We know that applying voting rules to profiles consisting of arbitrary preferences on multi-issue domains is computationally difficult. Does it become significantly easier when we restrict to conditionally lexicographic preferences? The question, of course, may depend on the voting rule used.

A *conditionally lexicographic profile* is a collection of n conditionally lexicographic preferences over $\mathcal{X}$. As conditionally lexicographic preferences are compactly represented by LP-trees, we define a *LP-profile* P as a collection of n LP-trees. Given a class $\mathcal{C}$ of LP-trees, let us call $\mathcal{C}$*-profile* a finite collection of LP-trees in $\mathcal{C}$.

Given a LP-profile P and a voting rule r, a naive way of finding the co-winners would consists in determining the n linear orders induced by the LP-trees and then apply r to these linear orders. However, this would be very inefficient, both in space and time. We would like to know how feasible it is to compute the winners directly from the LP-trees. More specifically, we ask the following questions: (a) given a voting rule, how difficult is it to compute the co-winners (or, else, one of the co-winners) for the different classes of LP-trees? (b) for score-based rules, how difficult is it to compute the score of the co-winners? (c) is it possible to have, for some voting rules and classes of LP-trees, a compact representation of the set of co-winners?

Formally, we consider the following decision and function problems.

Definition 1 *Given a class $\mathcal{C}$ of LP-trees and a voting rule r that is the maximizer of scoring function S, in the S-SCORE and EVALUATION problems, we are given a $\mathcal{C}$-profile P and an alternative $\vec{d}$. In the S-SCORE problems, we are asked to compute whether $S(P, \vec{d}) > T$ for some given $T \in \mathbb{N}$. In the EVALUATION problem, we are asked to compute whether there exists an alternative $\vec{d}$ with $S(P, \vec{d}) > T$ for some given $T \in \mathbb{N}$. In the WINNER problem, we are asked to compute $r(P)$.*

When we say that WINNER for some voting rule w.r.t. some class $\mathcal{C}$ is in P, the set of winners can be compactly represented, and can be computed in polynomial time.

Note that if EVALUATION is NP-hard and the score of an alternative can be computed in polynomial time, then WINNER cannot be in P unless P = NP: if WINNER were in P, then EVALUATION could be solved in polynomial time by computing a winner and its score.

For the voting rules studied in this paper, if not mentioned specifically, EVALUATION is w.r.t. the score functions we present when defining these rules. In this paper, we only show hardness proofs, membership in NP or #P is straightforward.

3.2 Two Specific Cases: Fixed Importance and Fixed Preference

It is worth focusing on the specific case of the class of profiles composed of LP-trees which have a fixed, linear structure: there is an order of importance among issues, which is common to all voters: X_1 is more important than X_2, which is itself more important than X_3, and so on.... Voters of course may have differing local preferences for the value for each issue, and their preferences on each issue may depend on the values of more important issues. A simple, easy to compute, and cheap in terms of communication, rule works as follows [14]: choose a value for X_1 according to the majority rule (possibly with a tie-breaking mechanism if we have an even number of voters); then, choose a value for X_2 using again the majority rule; and so on. The winner is called the *sequential majority winner*. When there is an odd number of voters, the sequential majority winner is the Condorcet winner (cf. Proposition 3 in [14], generalized in [7] to CI-profiles in which all voters have the same importance tree.). This, together with the fact that the sequential majority winner can be computed in polynomial time, shows that the winner of any Condorcet-consistent rule applied to FI profiles can be computed in polynomial time.

The case of fixed preferences is very specific for a simple reason: in this case, the top-ranked alternative is the same for all voters! This makes the winner determination trivial for all reasonable voting rules. However, nontrivial problems arise if we have constraints that limit the set of feasible alternatives. We devote Section 7 to aggregating FP trees.

4 k-Approval

We start by the following lemma. Most proofs are omitted due to the space constraint.

Lemma 1 *Given a positive integer k' such that $1 \leq k' \leq 2^p$ written in binary, and an LP-tree $\mathcal{L}$, the k'-th preferred alternative of $\succ_{\mathcal{L}}$ can be computed in time $O(p)$ by Algorithm 1.*

Algorithm 1: $FindAlternative(\mathcal{L}, k')$

Let $k^* = (k^*_{p-1}...k^*_0)_2 = 2^p - k'$ and $\mathcal{L}^* = \mathcal{L}$;
for $i = p - 1$ *down to* $i = 0$ **do**
  Let X_j be the root issue of $\mathcal{L}^*$ with local preferences $x_j \succ \overline{x_j}$;
  if $k^*_i = 1$ **then**
    Let $\mathcal{L}^* \leftarrow \mathcal{L}^*(x_j)$ (the subtree of $\mathcal{L}^*$ tracing the path $X_j = x_j$) and let $a_j = x_j$;
  end
  else Let $\mathcal{L}^* \leftarrow \mathcal{L}^*(\overline{x_j})$ and let $a_j = \overline{x_j}$;
end
return $\vec{a}$.

Similarly, the position of a given alternative $\vec{d}$ can be computed in time $O(p)$. It follows that the k-approval score of any alternative in a CI-CP profile can be computed in time $O(np)$. However, this does not mean that the winner can be computed easily, because the number of alternatives is exponential in p. For some specific values of k, though, computing the k-approval winner is in P.

Proposition 1 *Let k be a constant independent of p. When the profile is composed of n LP-trees, computing the k-approval co-winners for P can be done in time $O(knp)$.*

Proof: We compute the top k alternatives of each LP-tree in P; we store them in a table together with their k-approval score. As we have at most kn such alternatives, constructing the table takes $O(knp)$. □

A similar result also holds for computing the $(2^p - k)$-approval co-winners for any constant k.[2]

Theorem 1 (CI-CP) *For CI-CP profiles,* WINNER *for 2^{p-1}-approval can be computed in time $O(np)$.*

Proof: We note that an alternative $\vec{d}$ is among the first half of alternatives in $\mathcal{L}_j$ *iff* the root issue of $\mathcal{L}_j$ is assigned to the preferred value. We build a table with the following $2p$ entries $\{1_1, 0_1, \ldots, 1_p, 0_p\}$: for every $\mathcal{L}_j$ we add 1 to the score of 1_i (resp. 0_i) if X_i is the root issue of $\mathcal{L}_j$ and the preferred value is 1_i (resp. 0_i). When this is done, for each X_i, we instantiate X_i to 1_i (resp. 0_i) if the score of 1_i is larger than the score of 0_i (resp. vice versa); if the scores are identical, we do not instantiate X_i. We end up with a partial instantiation, whose set of models (satisfying valuations) is exactly the set of co-winners. □

Applying 2^{p-1}-approval here is both intuitive and cheap in communication (each voter only communicates her most important issue and its preferred value), and the output is computed very easily. On the other hand, it uses a very small part of the LP-trees. We may want to do better and take, say, the most important two issues into account, which comes down to using 2^{k-2}-approval or $(2^{k-1} + 2^{k-2})$-approval. However, this comes with a complexity cost. Let M be a constant independent of p and n and define $N(M, p)$ to be the set of all multiples of 2^{p-M} that are $\leq 2^p$, except 2^{p-1}. For instance, if $M = 3$ then $N(3, p) = \{2^{p-3}, 2^{p-2}, 2^{p-2}+2^{p-3}, 2^{p-1}+2^{p-3}, 2^{p-1}+2^{p-2}, 2^{p-1}+2^{p-2}+2^{p-3}\}$.

[2]However, there is little practical interest in using $2^p - k$ approval for a fixed (small) value of k, since in practice, we will have $kn \ll 2^p$, and almost every alternative will be a co-winner.

Theorem 2 (UI-UP) *For any $k \in N(M,p)$, for UI-UP profiles,* EVALUATION *for k-approval is* NP*-hard.*

Proof sketch: When $k = 2^{p-i}$ for some $i \geq 2$, the hardness of EVALUATION is proved by a reduction from the NP-complete problem MIN2SAT [12], where we are given a set Φ of clauses, each of which is the disjunction of two literals, and an integer T'. We are asked whether there exists a valuation that satisfy smaller than T' clauses in Φ. We next show the case $k = 2^{p-2}$ as an example. We note that $\vec{d}$ is among the first quarter of alternatives in $\mathcal{L}_j$ *iff* the root issue of $\mathcal{L}_j$ is assigned to the preferred value, and the second most important issue in $\text{IO}(\mathcal{L}_j, \vec{d})$ is assigned to the preferred value as well. Now, we give a polynomial reduction from MIN2SAT to our problem: given a set Φ of 2-clauses, the negation $\neg C_i$ of each clause $C_i \in \Phi$ is mapped into a UI-UP LP-tree whose top quarter of alternatives satisfies $\neg C_i$ (for instance, $\neg X_3 \wedge X_4$ is mapped into a LP-tree whose two most important issues are X_3 and X_4, and their preferred values are 0_3 and 1_4). The set of co-winners is exactly the set of valuations satisfying a maximal number of clauses $\neg C_i$, or equivalently, satisfying a minimal number of clauses in Φ.

The hardness for any other k in $N(M,p)$ is proved by a reduction from special cases of the MAXSAT problem, which are omitted due to the space constraint. □

The hardness proofs carry over to more general models, namely {UI,CI}×{UP,CP}. We next present an algorithm that converts winner determination for k-approval to a compact GENERALISED MAXSAT problem ("generalised" here means that the input is a set of formulas, and not necessarily clauses). The idea is, for each LP-tree $\mathcal{L}_j$, we construct a formula φ_j such that an alternative (valuation) is ranked within top k positions *iff* it satisfies φ_j. φ_j is further composed of the disjunction of multiple sub-formulas, each of which encodes a path from the root to a leaf in the tree structure, and the valuations that are ranked among top k positions.

Formally, for each path $\mathbf{u}$, we define a formula $C_{\mathbf{u}}$ that is the conjunction of literals, where there is an literal X_i (resp., $\neg X_i$) if and only if along the path $\mathbf{u}$, there is an edge marked 1_i (resp., 0_i). For any path with importance order $\mathcal{O}$ (w.l.o.g. $\mathcal{O} = X_1 \rhd X_2 \rhd \cdots \rhd X_p$) and $k = (k_{p-1} \ldots k_0)_2$ in binary, we define a formula $D_{\mathcal{O},k}$. Due to the space constraint, we only present the construction for the CI-UP case, but it can be easily extended to the CI-CP case. For each $i \leq p-1$, let $l_i = X_i$ if $1_i \succ 0_i$, and $l_i = \neg X_i$ if $0_i \succ 1_i$. Let $D_{\mathcal{O},k}$ be the disjunction of the following formulas: for every $i^* \leq p-1$ such that $k_{i^*} = 1$, there is a formula $(\bigwedge_{i>i^*:k_i=0} l_i) \wedge l_{i^*}$. To summarize, for each LP-tree $\mathcal{L}_j$ in the profile we have a formula φ_j, and we can use a (generalised) MAXSAT solver to find a valuation that maximizes the number of satisfied formulas $\{\varphi_j\}$. Note that there are efficient such solvers; see, e.g., [16] and the *Minimally Unsatisfiable Subset Track* of the 2011 Sat Competition, at http://www.satcompetition.org/2011/#tracks.

Example 2 *Let $\mathcal{L}$ denote the LP-tree in Example 1, except that the preferences for t is unconditionally $0_2 \succ 1_2$. Let $k = 5 = (101)_2$. For the upper path we have the following clause $(\neg X_1) \wedge (\neg X_1 \vee (\neg X_2 \wedge X_3))$. For the lower path we have the following formula $(X_1) \wedge (\neg X_1 \vee (X_3 \wedge \neg X_2))$.*

Theorem 3 *For any $k \leq 2^p - 1$ represented in binary and any profile P of LP-trees, there is a polynomial-size set of formulas Φ such that the set of k-approval co-winners for P is exactly the set of the models of* MAXSAT*(Φ).*

Therefore, though WINNER for k-approval is hard to compute for some cases, it can be done efficiently in practice by using a generalized MAXSAT solver.

Note that all polynomiality results for k-approval carry on to the *Bucklin* voting rule (that we do not recall): it suffices to apply k-approval dichotomously until we get the value of k for which the score of the winner is more than $\frac{n}{2}$.

Now, we focus on the specific case of fixed importance orders (FI).

Theorem 4 (FI-CP) *Let $k \in N(M,p)$. For FI-CP profiles,* WINNER *for k-approval can be computed in time $O(2^M \cdot n)$.*

Proof sketch: For simplicity, we only present the algorithm for the case $k = 2^{p-2}$. The other cases are similar. Let $X_1 > X_2 > \dots$ be the importance order, common to all voters. There are four types of votes: those for which the 2^{p-2} top alternatives are those satisfying $\gamma_1 = X_1 \wedge X_2$ (type 1), those satisfying $\gamma_2 = X_1 \wedge \neg X_2$ (type 2), etc. Let α_i be the number of votes in P of type i ($i = 1, 2, 3, 4$). The 2^{p-2}-approval co-winners are the alternatives that satisfy γ_i such that $\alpha_i = \max\{\alpha_i, i = 1, \dots, 4\}$. □

5 Borda

We start with a lemma that provides a convenient localized way to compute the Borda score for a given alternative in an LP-tree $\mathcal{L}$. For any $\vec{d} = (d_1, \dots, d_p) \in \mathcal{X}$ and any $i \leq p$, we define the following notation, which is an indicator whether the i-th component of $\vec{d}$ is preferred to its negation in $\mathcal{L}$, given the rest of values in $\vec{d}$, denoted by $\vec{d}_{-i}$.

$$\Delta_i(\mathcal{L}, \vec{d}) = \begin{cases} 1 & \text{if in } \mathcal{L}, d_i \succ \overline{d_i} \text{ given } \vec{d}_{-i} \\ 0 & \text{Otherwise} \end{cases}$$

$\Delta_i(\mathcal{L}, \vec{d})$ can be computed in polynomial-time by querying the CPT of X_i along $\mathrm{IO}(\mathcal{L}, \vec{d})$. We let $\mathrm{rank}(X_i, \mathcal{L}, \vec{d})$ denote the rank of issue X_i in $\mathrm{IO}(\mathcal{L}, \vec{d})$.

Lemma 2 *For any LP-tree $\mathcal{L}$ and any alternative $\vec{d}$, we have the following calculation:*

$$S_{Borda}(\mathcal{L}, \vec{d}) = \sum_{i=1}^{p} 2^{p - rank(X_i, \mathcal{L}, \vec{d})} \cdot \Delta_i(\mathcal{L}, \vec{d})$$

Example 3 *Let $\mathcal{L}$ denote the LP-tree defined in Example 1. We have $S_{Borda}(\mathcal{L}, 011) = 2^2 \cdot 1 + 2^1 \cdot 0 + 2^0 \cdot 1 = 5$ and $S_{Borda}(\mathcal{L}, 101) = 2^2 \cdot 0 + 2^0 \cdot 0 + 2^1 \cdot 1 = 2$.*

Hence, the Borda score of $\vec{d}$ for profile $P = (\mathcal{L}_1, \dots, \mathcal{L}_n)$ is $S_{\text{Borda}}(P, \vec{d}) = \sum_{j=1}^{n} \sum_{i=1}^{p} 2^{p - \mathrm{rank}(X_i, \mathcal{L}_j, \vec{d})} \cdot \Delta_i(\mathcal{L}_j, \vec{d})$.

Theorem 5 (CI-UP) *For CI-UP profiles,* EVALUATION *is* NP*-hard for Borda.*

Proof sketch: We prove the NP-hardness by a reduction from 3SAT. Given a 3SAT instance, we construct an EVALUATION instance, where there are $q + 2$ issues $\mathcal{I} = \{c, d\} \cup \{X_1, \dots, X_q\}$. The clauses are encoded in the following LP-trees: for each $j \leq t$, we define an LP-tree $\mathcal{L}_j$ with the following structure. Suppose C_j contains variables $X_{i_1}, X_{i_2}, X_{i_3}$ ($i_1 < i_2 < i_3$), and $d_{i_1}, d_{i_2}, d_{i_3}$ are the valuations of the three variables that satisfy C_j. In the importance order of $\mathcal{L}_j$, the first three issues are $X_{i_1}, X_{i_2}, X_{i_3}$. The fourth issue is c and the fifth issue is d if and only if $X_{i_1} = d_{i_1}$, $X_{i_2} = d_{i_2}$, or $X_{i_2} = d_{i_2}$; otherwise the fourth issue is d and the fifth issue is c. The rest of issues are ranked in the alphabetical order (issues in C are ranked higher than issues in $\mathcal{S}$). Then, we set the threshold appropriately (details omitted due to the space constraint) such that the Borda score of an alternative is higher than the threshold if and only if its d-component is 1, and the its values for $X_1, \dots, X_p$ satisfy all clauses. □

Finally, we show that WINNER for Borda can be converted to a weighted generalized MAXSAT problem. We note that $\Delta_i(\mathcal{L}_j, \vec{d})$ can be represented compactly by a formula φ_j^i such that a valuation $\vec{d}$ satisfies φ_j^i *iff* $\Delta_i(\mathcal{L}_j, \vec{d}) = 1$. The idea is similar to the logical formula for k-approval, where each path $\mathbf{u}$ corresponds to a clause $C_{\mathbf{u}}$, and there is another clause depicting whether $\Delta_i(\mathcal{L}_j, \vec{d}) = 1$ in $\mathbf{u}$. For example, let $\mathcal{L}$ denote the LP-tree in Example 1, then $\Delta_2(\mathcal{L}, \vec{d})$ can be presented by the disjunction of the clauses for the two paths: $\neg X_1 \wedge \neg X_2$ for the upper path, and $X_1 \wedge ((\neg X_3 \wedge \neg X_2) \vee (X_3 \wedge X_2))$ for the lower path.

Theorem 6 *For any profile P of LP-trees, there is a set of clauses Φ with weights such that the set of Borda co-winners for P is exactly the set of the models of* WEIGHTED MAXSAT*(Φ).*

Now, we focus on the specific case of unconditional importance orders (UI). When, for each $\mathcal{L}_j$ the importance order is unconditional, $\text{rank}(X_i, \mathcal{L}_j, \vec{d})$ does not depend on $\vec{d}$: let us denote it $\text{rank}(X_i, \mathcal{L}_j)$. It can be computed in polynomial time by a simple exploration of the tree $\mathcal{L}_j$.

If the preferences are unconditional, then the Borda winner is the alternative $\vec{d}$ that maximises $\sum_{i=1}^{p} \sum_{j=1}^{n} 2^{p-\text{rank}(X_i,\mathcal{L}_j)} \Delta_i(\mathcal{L}_j, \vec{d})$. We can choose in polynomial time the winning value for each issue independently: it is the d_i that maximizes

$$\sum_{j=1}^{n} 2^{p-\text{rank}(X_i,\mathcal{L}_j)} \Delta_i(\mathcal{L}_j, d_i) \quad \text{where } \Delta_i(\mathcal{L}_j, d_i) = \begin{cases} 1 \text{ if in } \mathcal{L}_j, d_i \succ \overline{d_i} \\ 0 \text{ otherwise.} \end{cases}$$

Note that this method still works if the voters have differing importance order – provided they still have unconditional importance.

Theorem 7 (UI-UP) *For UI-UP profiles,* WINNER *for Borda can be computed in polynomial time.*

However, if we allow conditional preferences, computing the Borda winner becomes intractable:

Theorem 8 (FI-CP) *For FI-CP profiles,* EVALUATION *is* NP*-hard for Borda.*

6 Condorcet-Consistent Rules

We start by studying the several classes of conditionally lexicographic preferences according to the existence of a Condorcet winner. We recall the following result from [7]:

Lemma 3 *[7] For FI-CP profiles, there always exists a Condorcet winner, and it can be computed in polynomial time.*

Proposition 2 *The existence of a Condorcet winner for our classes of conditionally lexicographic preferences is depicted on the table below, where* yes *(resp.* no*) means that the existence of a Condorcet winner is guaranteed (resp. is not guaranteed) for an odd number of voters.*

	FP	*UP*	*CP*
FI	*yes*	*yes*	*yes*
UI	*yes*	*no*	*no*
CI	*yes*	*no*	*no*

Proof: We know from [7] that for FI-CP profiles, there always exists a Condorcet winner, and it can be computed in polynomial time. For CI-FP profiles, since all voters have the same top alternative, the existence of a Condorcet winner is trivial. Finally, here is a UI-UP profile with two variables and three voters, that has no Condorcet winner:
– Voter 1: $[X \rhd Y], x \succ \bar{x}, y \succ \bar{y}$, and the linear order is $[xy \succ x\bar{y} \succ \bar{x}y \succ \bar{x}\bar{y}]$.
– Voter 2: $[Y \rhd X], \bar{x} \succ x, y \succ \bar{y}$, and the linear order is $[\bar{x}y \succ xy \succ \bar{x}\bar{y} \succ x\bar{y}]$.
– Voter 3: $[Y \rhd X], \bar{x} \succ x, \bar{y} \succ y$, and the linear order is $[\bar{x}\bar{y} \succ x\bar{y} \succ \bar{x}y \succ xy]$. □

Theorem 9 (UI-UP) *For UI-UP profiles, deciding whether a given alternative is the Condorcet winner is* coNP*-hard.*

Corollary 1 *For UI-UP profiles,* EVALUATION *for maximin is* coNP*-hard.*

7 Fixed Preferences

When the agents' local preferences are fixed (w.l.o.g. $1 \succ 0$), issues can be seen as objects, and every agent has a preference for having an object rather than not, everything else being equal. Obviously, the best outcome for every agent is $\vec{1}$, and applying any reasonable voting rule (that is, any voting rule that satisfies *unanimity*) will select this alternative, making winner determination trivial. However, winner determination ceases to be trivial if we have constraints that limit the set of feasible alternatives. For instance, we may have a maximum number of objects that we can take.

Let us start with the only tractability result in this section, with the Borda rule. Recall that, when, for each $\mathcal{L}_j$ the importance order is unconditional, $\text{rank}(X_i, \mathcal{L}_j)$ does not depend on $\vec{d}$. If, the preferences are fixed, $\Delta_i(\mathcal{L}_j, \vec{d}) = d_i$, and $S_{\text{Borda}}(P, \vec{d}) = \sum_{i=1}^{p} d_i \sum_{j=1}^{n} 2^{p-\text{rank}(X_i,\mathcal{L}_j)}$. We have the following theorem, which states that for the UI-FP case, computing the Borda winner is equivalent to computing the winner for a profile composed of importance orders, by applying some positional scoring rule. For any order $\rhd$ over $\mathcal{I}$, let $ext(\rhd)$ denote the UI-FP LP-tree whose importance order is $\rhd$.

Theorem 10 (UI-FP) *Let f_p denote the positional scoring rule over $\mathcal{I}$ with the scoring vector $(2^{p-1}, 2^{p-2}, \ldots, 0)$. For any profile $P_{\mathcal{I}}$ over $\mathcal{I}$, we have $ext(f_p(P_{\mathcal{I}})) = Borda(ext(P_{\mathcal{I}}))$.*

However, when the importance order is conditional, the Borda rule becomes intractable. We prove that using the following problem:

Definition 2 *Let voting rule r be the maximizer of scoring function S. In the K-EVALUATION problem, we are given a profile P that is composed of lexicographic preferences whose local preferences for all issues are $1 \succ 0$, a natural number K, and an integer T. We are asked to compute whether there exists an alternative $\vec{d}$ that takes 1 on no more than K issues and $S(P, \vec{d}) > T$.*

Theorem 11 (CI-FP) *For CI-FP profiles, K-EVALUATION is NP-hard for Borda.*

Theorem 12 (UI-FP) *Let $k \in N(M, p)$. For UI-FP profiles,* K-EVALUATION *for k-approval is NP-hard.*

Theorem 13 (UI-FP) *For UI-FP profiles, Copeland-SCORE is #P-hard.*

The proof is by polynomial-time counting reduction from #INDEPENDENT SET. Maximin, when the preferences are fixed (to be $1 \succ 0$ for all issues), the maximin score of $\vec{1}$ is 0 and the maximin score of any other alternative is $2^p - 1$. This trivialize the computational problem of winner determination even when with the restriction on the number of issues that take 1 (if $K \neq p$ then all available alternatives are tied). Following Lemma 3, for FI profiles, the winner can be computed in polynomial-time.

8 Summary and Future Work

Our main results are summarized in Table 1. In addition, we can also show that for k-approval (except $k = 2^{p-1}$), Copeland and maximin, there is no observation similar to Theorem 10, and the maximin score of a given alternative is APX-hard to approximate.

Our conclusions are partly positive, partly negative. On the one hand, there are voting rules for which the domain restriction to conditionally lexicographic preferences brings significant benefits: this is the case, at least, for k-approval for some values of k. The Borda rule can be applied easily provided that neither the importance relation and the local preference are unconditional, which is a very strong restriction. The hardness of checking whether an alternative is a Condorcet winner suggest that Condorcet-consistent rules appears to be hard to apply as well. However, as we have

<table>
<tr><th></th><th>FP</th><th>UP</th><th>CP</th></tr>
<tr><td>FI</td><td>Trivial</td><td colspan="2">P
(Thm. 4)</td></tr>
<tr><td>UI</td><td rowspan="2">NPC
(Thm. 12)</td><td colspan="2" rowspan="2">NPC
(Thm. 2)</td></tr>
<tr><td>CI</td></tr>
</table>

(a) k-approval, $k \in N(M, p)$.

<table>
<tr><th></th><th>FP</th><th>UP</th><th>CP</th></tr>
<tr><td>FI</td><td>Trivial</td><td rowspan="2">P
(Thm. 7)</td><td rowspan="3">NPC
(Thm. 8)</td></tr>
<tr><td>UI</td><td>P</td></tr>
<tr><td>CI</td><td>NPC
(Thm. 11)</td><td>NPC
(Thm. 5)</td></tr>
</table>

(b) Borda.

<table>
<tr><th></th><th>FP</th><th>UP</th><th>CP</th></tr>
<tr><td>FI</td><td>Trivial</td><td colspan="2">Polynomial
(Lemma 3)</td></tr>
<tr><td>UI</td><td colspan="3" rowspan="2">#P-complete
(Thm. 13)</td></tr>
<tr><td>CI</td></tr>
</table>

(c) Copeland score.

<table>
<tr><th></th><th>FP</th><th>UP</th><th>CP</th></tr>
<tr><td>FI</td><td rowspan="3">Trivial</td><td colspan="2">P
(Lemma 3)</td></tr>
<tr><td>UI</td><td colspan="2" rowspan="2">coNPC
(Thm. 9, Coro. 1)</td></tr>
<tr><td>CI</td></tr>
</table>

(d) Maximin and Condorcet winner.

Table 1: Summary of computational complexity results.

shown that some of these problems can be reduced to a compact MAXSAT problem. From a practical point of view, it is important to test the performance of MAXSAT solvers on these problems. We believe that continuing studying preference representation and aggregation on combinatorial domains, taking advantages of developments in efficient CSP techniques, is a promising future work direction.

Acknowledgments

We thank all anonymous reviewers of AAAI-12, CP-12, and COMSOC-12 for their helpful comments and suggestions. This work has been partly supported by the project ComSoc (ANR-09-BLAN-0305-01). Lirong Xia is supported by NSF under Grant #1136996 to the Computing Research Association for the CIFellows Project.

References

[1] M. Ahlert. Aggregation of lexicographic orderings. *Homo Oeconomicus*, 25(3/4):301–317, 2008.

[2] S. Airiau, U. Endriss, U. Grandi, D. Porello, and J. Uckelman. Aggregating dependency graphs into voting agendas in multi-issue elections. In *Proceedings of IJCAI-11*, pages 18–23, 2011.

[3] R. Booth, Y. Chevaleyre, J. Lang, J. Mengin, and C. Sombattheera. Learning conditionally lexicographic preference relations. In *Proceeding of ECAI-10*, pages 269–274, 2010.

[4] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus statements. *Journal of Artificial Intelligence Research (JAIR)*, 21:135–191, 2004.

[5] R. I. Brafman, C. Domshlak, and S. E. Shimony. On graphical modeling of preference and importance. *Journal of Artificial Intelligence Research (JAIR)*, 25:389–424, 2006.

[6] S. Brams, D. Kilgour, and W. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15(2):211–236, 1998.

[7] V. Conitzer and L. Xia. Approximating common voting rules by sequential voting in multi-issue domains. In *Proceedings of KR-12*, 2012.

[8] J. Dombi, C. Imreh, and N. Vincze. Learning lexicographic orders. *European Journal of Operational Research*, 183:748–756, 2007.

[9] M. Garey and D. Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[10] G. Gigerenzer and D. Goldstein. Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review*, 103(4):650–669, 1996.

[11] B. Jaumard and B. Simeone. On the complexity of the maximum satisfiability problem for horn formulas. *Information Processing Letters*, 26(1):1–4, 1987.

[12] R. Kohli, R. Krishnamurti, and P. Mirchandani. The minimum satisfiability problem. *SIAM Journal on Discrete Mathematics*, 7(2):275–283, 1994.

[13] D. Lacy and E. M. Niou. A problem with referendums. *Journal of Theoretical Politics*, 12(1):5–31, 2000.

[14] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57(3):304–324, 2009.

[15] G. D. Pozza, M. S. Pini, F. Rossi, and K. B. Venable. Multi-agent soft constraint aggregation via sequential voting. In *Proceedings of IJCAI-11*, pages 172–177, 2011.

[16] V. Ryvchin and O. Strichman. Faster extraction of high-level minimal unsatisfiable cores. In *Theory and Applications of Satisfiability Testing - SAT 2011*, volume 6695 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 2011.

[17] M. Schmitt and L. Martignon. On the complexity of learning lexicographic strategies. *Journal of Machine Learning Research*, 7:55–83., 2006.

[18] F. Yaman, T. Walsh, M. Littman, and M. desJardins. Democratic approximation of lexicographic preference models. In *Proceedings of ICML-08*, pages 1200–1207, 2008.

Jérôme Lang
LAMSADE
Université Paris-Dauphine
France
Email: `lang@lamsade.dauphine.fr`

Jérôme Mengin
IRIT
Université de Toulouse
France
Email: `mengin@irit.fr`

Lirong Xia
School of Engineering and Applied Sciences
Harvard University
USA
Email: `lxia@seas.harvard.edu`

Online (Budgeted) Social Choice

Brendan Lucier, Joel Oren

Abstract

We consider classic social choice problems in an online setting. In the problems we consider, a decision-maker must select a subset of candidates in accordance to reported preferences, e.g. to maximize the value of a scoring rule. However, agent preferences cannot be accessed directly; rather, agents arrive one at a time to report their preferences, and each agent cares only about those candidates that have been selected by the time she arrives. On each step, the decision maker must choose whether to irrevocably add candidates to the final selection set given the preferences observed so far, with the goal of maximizing the average score over all agents.

We show that when preferences are arbitrary but agents arrive in uniformly random order, an online selection algorithm can approximate the optimal value of an arbitrary positional scoring function to within a factor of $(1-1/e)-o(1)$ as the number of agents grows large, nearly matching the performance of the best offline polynomial-time algorithm. When agent preferences are drawn from a Mallow's model distribution, a different selection algorithm achieves approximation factor that limits to 1 as the number of agents grows large. Our methods are straightforward to implement, and draw upon connections to online computation and secretary problems.

1 Introduction

Suppose that a manufacturer wishes to focus on a selected set of possible products to offer to incoming consumers. On each day a new client arrives, selecting her favorite product among those being offered. However, the client may also express preferences over *potential* products, including those that are not currently being offered. The manufacturer must then decide whether or not to add new production lines to make available to that consumer (as well as to future consumers). While adding a new product would potentially increase customer welfare, it carries with it some opportunity cost: it would be impractical to offer every possible product, so choices are effectively limited and irrevocable (since new production lines incur substantial overhead). Adding new products may be worthwhile if many future customers would prefer the chosen product as well, though this is not known to the manufacturer in advance. The problem is thus one of online decision-making, where uncertainty of future preferences must be balanced with the necessity of making decisions to realize current gains.

Such a setting gives rise to obvious complications. On one hand, adding an item that is highly ranked by the current user to the list of available items will satisfy the current client. On the other hand, in such settings there is usually an underlying constraint that prohibits the addition of arbitrarily many items. In our study of this problem, we will address settings in which the underlying restriction is a cardinality constraint, which limits the number of chosen alternatives. The main problem that we are facing is therefore an online social choice problem: we are required to choose the most "favourable" set of candidates, while having only a partial view of the objective function. For any given offline social choice problem, such as selecting a candidate to maximize the value of a certain scoring function of the user preferences, one might consider an online variant in which each agent receives value only for those candidates that have been selected at or before the time that he arrives. In this case, the objective function is the average of the agent scores, determined by a prescribed positional scoring function or hidden utility function.

We consider various different models for the manner in which the agents preferences are set. In the distributional model, the player preferences are drawn independently from a distribution over permutations. For example, one might assume that the preferences are sampled from a parametrized Mallows model, which defines a unimodal distribution over permutations. An alternative approach that is common to online algorithm analysis is to suppose that the set of agent preferences is set arbitrarily (i.e. adversarially), but that the order of agent arrival is random[1]. Utilizing previous results in the area of online matching, we show that our methods for this adversarial setting carry over to the case in which the preferences are drawn independently from an *unknown* distribution.

As previously mentioned, our ultimate goal is to maximize the average score of the agents when each agent is matched with his most preferred item available at the time of arrival. Generally speaking, our finding is that if the number of agents is sufficiently large compared to the number of candidates, it is possible to design online algorithms that perform asymptotically as well as the best possible offline algorithms, with high probability. Our approach is reminiscent of those used for well-studied secretary-type problems, in which the candidates arrive online rather than the agents. Our results also suggest a number of potential extensions for future research, which we discuss in our concluding remarks.

Results We first consider adversarial settings, where agent preferences are arbitrary but arrive in uniformly random order. We show that one can approximate the optimal choice of a *single* candidate, with respect to an *arbitrary* positional scoring function, with approximation ratio $(1 - o(1))$ where the asymptotic notation is with respect to the number of agents. In other words, the regret exhibited by the online selection method vanishes as n grows large. If more than one alternative can be chosen, say $k > 1$ in total, we show that for any positional scoring function, combining our sample approach with a standard greedy algorithm for submodular set-function maximization provides a $(1 - \left(\frac{k-1}{k}\right)^k - o(1))$ approximation to the optimal choice. Thus, as n grows large, our online algorithm achieves approximation factor $1 - 1/e$, matching a lower bound for offline algorithms [12].

Moving away from positional scoring functions to arbitrary utility functions, we apply a recent result due to Boutilier et al. [3] who demonstrated that a social choice function can approximate the choice of a candidate to maximize agent utilities to within a factor of $\tilde{O}(\sqrt{m})$ (where m is the number of candidates), even if only preference lists are made available. Using our results for arbitrary positional scoring functions, we obtain similar bounds for the problem of maximizing average agent utility in an online fashion, with vanishing additional errors due to sampling.

Finally, in the distributional setting where preferences are drawn from a parameterized Mallows model, we show that for the selection of $k \geq 1$ alternatives under an arbitrary positional scoring rule, one can obtain an approximation ratio of $(1 - o(1))$, suffering regret that vanishes as n grows. In the particular case of Borda ranking, we show that sampling a logarithmic number of agents is sufficient for approximating the optimal k-set.

2 Related Work

The problem of selecting a single candidate given a sequence of agent preference lists is the traditional social choice problem. The budgeted form of this offline problem was introduced by Chamberlin and Courant [5], and subsequently studied by Boutilier and Lu [12], in which several natural constraints on the allocated set were considered. In particular, it is shown that for the case where producing copies of the alternatives bears no cost, the problem of selecting which candidates to make available is a straightforward case of non-decreasing and

[1]For the problems we consider, as with many others, no algorithm can guarantee reasonable performance if the adversary is also allowed to set the order of the arrival of agents.

submodular set-function maximization, subject to a cardinality constraint, which admits a simple greedy algorithm with approximation ratio $1 - 1/e$. Our work differs in that the agent preferences arrive online, complicating the choice of which alternatives to select, as the complete set of agents preference is not fully known in advance.

In our online setting, we refer to the Mallows model ([14]), a well-studied model for distributions over permutations (e.g. [8, 6]) which has been studied and extended in various ways. In recent work, Braverman and Mossel have shown that the sample complexity required to estimate the maximum-likelihood ordering of a given Mallows model distribution is roughly linear [4]. We make use of some of their results in our analysis.

Adversarial and stochastic analysis in online computation have received considerable attention (e.g. [7]). In our analysis, we make critical use of the assumption that agent arrivals are randomly permuted. This is a common assumption in online algorithms (e.g. [10, 11, 13]). Correspondingly, in our analysis of the adversarial model, we use techniques that resemble methods used in secretary and multi-armed bandits problems (see [2] for a survey), of partially observing some initial data, and bounding the total error.

A recent paper by Boutilier et al. [3] considered the social choice problem from a utilitarian perspective, where agents have underlying utility functions that induce their reported preferences. The authors introduce a measure of *distortion* to compare the performance of their social choice functions to the social welfare of the optimal alternative. We make use of their constructions in our results for the utilitarian model.

The online arrival of preferences has been previously studied by Tennenholtz [16]. This work postulates a set of voting rule axioms that are compatible with online settings.

3 Preliminaries

Given is a ground set of alternatives (candidates) $A = \{a_1, \ldots, a_m\}$. An agent $i \in N = \{1, \ldots, n\}$, has a preference $\succ_i$ over the alternatives, represented by a permutation π^i. For a permutation π and an alternative $a \in A$, we will let $\pi(a)$ denote the rank of a in π. A *positional scoring function* (PSF) assigns a score v_i to the alternative ranked ith, given a prescribed vector $\mathbf{v} \in \mathbb{R}^m_{\geq 0}$. Given an (implicit) set of agent preferences, we will denote the average score of a *single* element $a \in A$ by $\overline{F}(a) = \frac{1}{n}\sum_{i=1}^n F_i(a)$, where $F_i(a) = \mathbf{v}(\pi^i(a))$. Moreover, we will consider the score of a *set* $S \subseteq A$ of candidates w.r.t. to a set of agents as the average positional scores of each of the agents, assuming that each of them selected their highest ranked candidate in the set: $\overline{F}(S) = \frac{1}{n}\sum_{i \in N} \max_{a \in S} F_i(a)$.

The online budgeted social choice problem. We consider the problem of choosing a set of $k \geq 1$ candidates from the set of potential alternatives. An algorithm for this problem starts with an empty "slate" $S_0 = \emptyset$ of alternatives, of prescribed capacity $k \leq m$. In each step $t \in [n]$, an agent arrives and reveals her preference ranking. Given this, the algorithm can either add new candidates $I \subseteq A \setminus S_{t-1}$ to the slate (i.e. set $S_t \leftarrow S_{t-1} \cup I$), if $|S_{t-1}| + |I| \leq k$, or leave it unchanged. Agent i in turn takes a copy of one of the alternatives *currently* on the slate, i.e. S_t. Any addition of alternatives to the slate is *irrevocable*: once an alternative is added, it cannot be removed or replaced by another alternative. The offline version of this problem is called the limited choice model in [12].

Some of our results will make use of algorithms for maximizing non-decreasing submodular set functions subject to a cardinality constraint. A submodular set function $f : 2^U \to \mathbb{R}_{\geq}$ upholds $f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$ for all $S \subseteq T \subseteq U$ and $x \in U \setminus T$.

4 The Adversarial Model

We begin by supposing that the set of agent preference profiles is arbitrary, as might be chosen by an adversary. After the collection of all preference profiles has been fixed, we assume that they are presented to an online algorithm in a uniformly random order. The algorithm can irrevocably choose up to k candidates during any step of this process; each arriving candidate will then receive value corresponding to his most-prefered candidate that has already been chosen. The goal is to maximize the value obtained by the algorithm, with respect to an arbitrary positional scoring function[2].

In general, we cannot hope to achieve an arbitrarily close approximation factor to the optimal (in hindsight) choice of k candidates, as it is **NP**-hard to obtain better than a $(1 - \frac{1}{e})$ approximation to this problem even when all profiles are known in advance[3]. Our goal, then, is to provide an algorithm for which the approximation factor approaches $1 - \frac{1}{e}$ as n grows, matching the performance of the best-possible algorithm for the offline problem[4].

Let $F(\cdot)$ be an arbitrary PSF; without loss of generality we can scale F so that $F(1) = 1$. Note that this implies that $F(a) \in [0, 1]$ for each outcome a. If agent i has preference permutation π^i, then we write $F_i(\cdot) = F(\pi^i(\cdot))$ for the scoring function F applied to agent i's permutation of the choices. Also, we will write σ for the permutation of players representing the order in which they are presented to an online algorithm. Thus, for example, $F_{\sigma(1)}(a)$ denotes the value that the first observed player has for object a.

Given a set S of objects and PSF F, we write $F(S) = \max_{a \in S} F(a)$ for the value of the highest-ranked object in S. Given a set T of players, $F_T(S) = \sum_{j \in T} F_j(S)$ is the total score held by the players in T for the objects in S. We also write $\overline{F}_T(S) = \frac{F_T(S)}{|T|}$ for the average score assigned to set S. Let $OPT = \max_{S \subseteq A, |S| \leq k} F_N(S)$ be the optimal outcome value.

Let us first describe a greedy social choice rule for the offline problem that achieves approximation factor $(1 - 1/e)$, due to [12]. This algorithm proceeds by repeatedly selecting the candidate that maximizes the marginal gain in the objective value, until a total of k candidates have been chosen. As any PSF $F(\cdot)$ can be shown to be a (non-decreasing) submodular set-function over the sets of candidates (see for example, [12]), such an algorithm obtains approximation $1 - (\frac{k-1}{k})^k$, which is at most $1 - 1/e$ for all k. We will write $Greedy(N, k)$ for this algorithm applied to set of players N with cardinality bound k.

We now consider the online algorithm $\mathcal{A}$, listed as Algorithm 1 below.

Algorithm 1: Online Candidate Selection Algorithm

Input: Candidate set A, parameters k and n, online sequence of preference profiles

Let $t \leftarrow n^{2/3}(\log n + k \log m)$;
Observe the first t agents, $T = \{\sigma(1), \ldots, \sigma(t)\}$;
$S \leftarrow Greedy(T, k)$;
Choose all candidates in S and let the process run to completion;

We write $V(\mathcal{A})$ for the value obtained by this algorithm. We claim that the expected value obtained by $\mathcal{A}$ will approximate the optimal offline solution.

[2] A stronger adversary would not only be able to set the preferences of the voters, but also their order, or even set preferences adaptively. However, it is not hard to see that in such cases no non-trivial bounds can be obtained, as the adversary can strategically cause the algorithm to exhaust its budget and then set the preferences to be the worst possible from that point onward.

[3] One can reduce Max-k-Coverage to the budgeted social choice problem for the special case of l-approval: the PSF in which the first l positions receive score 1, and others receive score 0.

[4] For the special case of the Borda scoring rule, it can be shown that the algorithm that simply select a random k-set obtains a $1-O(1/m)$-approximation to the offline problem. Furthermore, this algorithm can be derandomized using the method of conditional expectations. We omit the proof due to space considerations.

Theorem 1. *If $m < n^{1/3-\epsilon}$ for any $\epsilon > 0$, then $E[V(\mathcal{A})] \geq (1 - (\frac{k-1}{k})^k - o(1))OPT$.*

The first step in the proof of Theorem 1 is the following technical lemma, which states that the preferences of the first t players provide a good approximation to the (total) value of every set of candidates, with high probability.

Lemma 2. *$Pr[\exists S, |S| \leq k : |\overline{F}_T(S) - \overline{F}(S)| > n^{-1/3}] < \frac{2}{n}$, where the probability is taken over the order in which the agents arrive.*

Proof. Choose any set S with $|S| \leq k$. For each $j \in [t]$, let X_j be a random variable denoting the value $F_{\sigma(j)}(S)$. Note that $E[X_j] = \overline{F}(S)$ for all j, and that $\overline{F}_T(S) = \frac{1}{t}\sum X_j$. By the Hoeffding inequality (without replacement), for any $\epsilon > 0$, $Pr[|\overline{F}_T(S) - \overline{F}(S)| > \epsilon] < 2e^{-\epsilon^2 t}$. By the union bound over all S with $|S| \leq k$,

$$Pr[\exists S, |S| \leq k : |\overline{F}_T(S) - \overline{F}(S)| > \epsilon] < 2\sum_{\ell=1}^{k}\binom{m}{\ell}e^{-\epsilon^2 t} \leq 2m^k e^{-\epsilon^2 t}.$$

Setting $t = n^{2/3}(\log n + k \log m)$ and $\epsilon = n^{-1/3}$ then yields the desired result. □

With Lemma 2 in hand, we can complete the proof of Theorem 1 as follows. Since $F_T(S)$ approximates $F(S)$ well for every S, our approach will be to sample T, choose the (offline) optimal output set according to the preferences of T, then apply this choice to the remaining bidders. This generates two sources of error: the sampling error bounded in Lemma 2, and the loss due to not serving the agents in T. By setting the value of t judiciously, and noting that OPT cannot be very small (it must be at least $\frac{n}{m}$), one can show that the relative error vanishes as n grows large. The details appear in the full version of the paper.

One special case of note occurs when $k = 1$; that is, there is only a single candidate to be chosen. In this case, the regret experienced by our online algorithm vanishes as n grows.

Corollary 3. *If $k = 1$ and $m < n^{1/3-\epsilon}$ for any $\epsilon > 0$, then $E[V(\mathcal{A})] \geq (1 - o(1))OPT$.*

4.1 A Correspondence with the Unknown Distribution Model

We now note a correspondence between the random order model analyzed above and a model in which rankings are drawn from an underlying distribution over preferences. This observation was first made by Karande et al. ([9]) in the context of online bipartite matching. Suppose there is an underlying distribution $\mathcal{D}$ over the set of rankings over the alternatives A. For each player $i \in N$, suppose the ranking π^i for player i is sampled independently from $\mathcal{D}$.

The following result due to Karande et al. states that our algorithm for the adversarial model with random arrival order applies to this unknown-distribution setting as well.

Claim 4 ([9]). *Let $\mathcal{A}$ be an algorithm for the online social problem under the random order model that obtains a expected competitive ratio of α. Then $\mathcal{A}$ obtains an expected approximation ratio of at least α for the online social choice problem in the unknown distribution model. Furthermore, hardness results in the unknown distribution model hold in the random order model as well.*

This result implies that algorithm $\mathcal{A}$ achieves approximation factor $(1 - (\frac{k-1}{k})^k - o(1))$ to the social choice problem when preferences are drawn from an unknown underlying distribution, and that it is NP-hard to achieve an approximation factor better than $(1 - 1/e)$.

5 A Utilitarian Approach

In the previous section we considered the problem of maximizing the social value of a positional scoring function in an online setting. However, it may be more natural in some circumstances to assume that each agent assigns a non-negative utility to each candidate, even though these utilities are hidden and only the preference lists are revealed to a potential social choice function. In such settings, one would wish to choose candidates that maximize overall social welfare (i.e. sum of utilities), again in an online fashion. However, this goal is hindered by the fact that the utilities themselves are never made available to the algorithm. In this section we adapt a general technique due to Boutillier et al. [3] to show that our result for online PSF maximization extends to approximate online utility maximization.

We assume that each agent $i \in N$ has a latent utility function $u_i \colon A \to \mathbb{R}_{\geq 0}$. A utility function u_i induces a preference profile $\pi(u_i) = \pi^i$ such that $\pi^i(a) > \pi^i(a')$ precisely[5] when $u_i(a) \geq u_i(a')$. We let $\pi(\mathbf{u})$ denote the induced preference profile given a utility profile $\mathbf{u}$.

As in [3], we will assume that utilities can be normalized so that $\sum_{a \in A} u_i(a) = 1$ for each i. This assumption essentially states that each agent has the same total weight assigned to her candidate utilities. Note that without this assumption it would be impossible to approximate the optimal social welfare, since a single agent could have a single utility score that dominates all others, but an algorithm with access only to the preference profiles would have no awareness of this fact.

Intuitively, we would like to choose an alternative $a \in A$ that maximizes the (unknown) social welfare $sw(a, \mathbf{u}) = \sum_{i=1}^{n} u_i(a)$, based solely on the reported vote profile $\overrightarrow{\pi} = \overrightarrow{\pi}(\mathbf{u}) = (\pi^1, \ldots, \pi^n)$ induced by the utility profile. Of course, the preference profile $\overrightarrow{\pi}$ does not completely capture all of the information in the utility profile, and hence we should expect some loss.

Our hope will be to find a social choice rule f such that, if it were applied to the preference profile $\overrightarrow{\pi}$, it would return a candidate that approximately maximizes $sw(a, \mathbf{u})$. The *distortion* of f is the worst-case approximation factor incured when f is applied $\overrightarrow{\pi}(\mathbf{u})$. This notion of distortion was first formalized by Procaccia and Rosenschein in [15], and has been used in subsequent studies of the social choice problem with partial (or noisy) information about the underlying utilities (e.g. [3]). The formal definition is as follows.

Definition 5 (distortion)**.** *Let $\overrightarrow{\pi} \in S_m^n$ be a preference profile, and let $f : S_m^n \to A$ be a social choice function. The distortion of f is then given by*

$$dist(\overrightarrow{\pi}, f) = \sup_{\mathbf{u}:\pi(\mathbf{u}) = \overrightarrow{\pi}(\mathbf{u})} \frac{\max_{a \in A} sw(a, \mathbf{u})}{sw(f(\overrightarrow{\pi}), \mathbf{u})} \tag{5.1}$$

In [3], Boutilier et al. proposed a randomized social choice rule f with distortion $O(\sqrt{m \log m})$, and provided a corresponding lower bound of $\Omega(\sqrt{m})$. This rule f makes use of a positional scoring function $H(\cdot)$, that they refer to as the *harmonic scoring function.* In the harmonic scoring function, the score of a candidate ranked in position i is $H_i = 1/i$. Given preference profile $\overrightarrow{\pi}$, rule f either a) with probability $1/2$, chooses each candidate a with probability proportional to $H_N(a) = \sum_{i \in N} H(\pi^i(a))$, or b) with the remaining probability $1/2$, returns a uniformly random candidate.

We will make use of this social choice rule f to design an online algorithm achieving social welfare within a factor of $O(\sqrt{m \log m})$ of the optimal welfare. As before, we assume an adversarial setting: the collection of agent preferences can be arbitrary, but they are presented to the algorithm in an order determined by a (uniform) random permutation σ. Our algorithm $\mathcal{A}$ is described as Algorithm 2, below.

[5] In keeping with our simplifying assumption that preference profiles do not include indifference, we can assume that ties in utility are broken in some consistent manner.

Algorithm 2: Online Candidate Selection Algorithm for Utility Maximization

Input: Candidate set A, parameter n, sequence of preference profiles arriving online

Let $t \leftarrow n^{2/3} \log n$;
Observe the first t agents, $T = \{\sigma(1), \ldots, \sigma(t)\}$;
$a^* \leftarrow f(\pi^{\sigma(1)}, \ldots, \pi^{\sigma(t)})$;
Choose candidate a^* and let the process run to completion;

Given a particular utility profile $\mathbf{u}$ we will write $E[sw(\mathcal{A})]$ to denote the expected social welfare of the outcome returned by $\mathcal{A}$, given preference profile $\overrightarrow{\pi}(\mathbf{u})$, over permutations σ and randomness in $\mathcal{A}$. We will also write OPT for the optimal social welfare attainable for $\mathbf{u}$, i.e. $OPT = \max_{a \in A} \sum_i u_i(a)$.

Theorem 6. *Suppose $n > m^3$. Then for all $\mathbf{u}$, $E[sw(\mathcal{A})] \geq \frac{1}{O(\sqrt{m} \log m)} OPT$.*

The idea behind the proof of Theorem 6 is to note that the algorithm for offline utility maximization due to Boutilier et al. [3] works primarily by applying the low-distorition PSF f. However, our Theorem 1 implies that PSF value maximization can be approximated well by an online algorithm. We can therefore approximate the set that maximizes the (offline) value of f in the online setting. As long as the errors due to sampling and omitting the first t agents are not too large, this then implies an approximation to the utility-maximizing candidate set. The details of the proof appear in the full version of the paper.

6 The Distributional Model

We next suppose that agent preferences are distributed according to the well-studied Mallows model, which defines a family of permutation distributions. Roughly speaking, Mallow's model assumes that preferences are aligned according to some base permutation $\hat{\pi}$, but each agent's permutation is (independently) perturbed according to a particular error measure. We begin by giving a formal definition of this distribution.

Let us begin our formal definition by introducing the Kendall-tau distance (which is also known as the Kemeny distance or the bubble-sort distance):

Definition 7 (Kendall-tau distance)**.** *For all $\pi, \pi' \in S_m$, the* Kendall-tau distance *between π and π' is $d_K(\pi, \pi') = \#\{i \neq j : \pi(i) < \pi(j) \text{ and } \pi'(i) > \pi'(j)\}$.*

Definition 8 (The Mallows model)**.** *Let $\phi \in (0, 1)$ and $\hat{\pi} \in S_m$. The Mallows model distribution $D(\hat{\pi}, \phi)$ is a distribution over permutations of $\{1, \ldots, m\}$, such that the probability of a permutation $\pi \in S_m$ is*

$$Pr[\pi] = \phi^{d_K(\pi, \hat{\pi})}/Z \tag{6.1}$$

where Z is a normalization constant: $Z = \sum_{\pi \in S_m} \phi^{d_K(\hat{\pi}, \pi)}$.

Fact 9. *It can be shown that $Z = 1 \cdot (1 + \phi) \cdot \cdots \cdot (1 + \cdots + \phi^{m-1})$.*

We note that the Mallows model induces a unimodal distribution. Furthermore, the parameter ϕ can be seen as controlling the amplitude of error with respect to permutation $\hat{\pi}$: as ϕ approaches 1 the distribution tends to uniformity, and as ϕ approaches 0 the distribution approaches a point mass at $\hat{\pi}$.

We will assume that the agent preference rankings are drawn independently from a Mallows model distribution $D(\hat{\pi}, \phi)$, where the underlying reference ranking $\hat{\pi}$ is unknown.

We will assume that the *dispersion parameter* ϕ is known in advance. Our optimization task in this model is to select a $S \subseteq A$ of size at most k, in an online fashion, so as to maximize the expected value of S among the remaining agents (with respect to a given positional scoring function).

For simplicity of notation and without loss of generality, from hereon we assume that $\hat{\pi}$ is the identity permutation. That is, $\hat{\pi}(i) = i$. We note that since $D(\hat{\pi}, \phi)$ is a unimodal distribution, Theorem 1 and Claim 4 together imply an immediate corollary for this distributional model.

Theorem 10. *Let $F(\cdot)$ be an arbitrary positional scoring function, and let $\mathcal{A}$ be the online algorithm listed as Algorithm 1. Then if $m < n^{1/3-\epsilon}$ for any $\epsilon > 0$, we have $E[V(\mathcal{A})] \geq (1 - (\frac{k-1}{k})^k - o(1))OPT$.*

Given this result, our motivating question for this section is whether we can obtain improved results by making use of the particular form of the Mallows model.

6.1 An Improved Result for Arbitrary PSFs

Suppose that our goal is to maximize the value of an arbitrary PSF $F(\cdot)$, scaled so that $F(1) = 1$. Write $A_m = \sum_{i=0}^{m-1} \phi^i$. We begin with a lemma about the Mallows model, which shows that in a sampled permutation π, we do not expect any particular candidate to be placed very far from its position in the reference ranking (the proof appears in the appendix of the full version paper):

Lemma 11. *Let $\pi \sim D(\hat{\pi}, \phi)$. Then for any $i \neq j$, $Pr[\pi^{-1}(i) = i] \geq Pr[\pi^{-1}(i) = j] + \frac{1-\phi}{A_m}$.*

Given this lemma, our strategy will be to observe many samples from the distribution, then attempt to guess the identities of the top k elements in the underlying permutation $\hat{\pi}$. Since each candidate is most likely to appear in its position from $\hat{\pi}$, we expect to be able to determine $\hat{\pi}$ after a relatively small number of samples. Our algorithm is provided as Algorithm 3, below.

Algorithm 3: Online Candidate Selection Algorithm for the Mallows Model

Input: Candidate set A, Mallows model parameter ϕ, parameter n, sequence of preference profiles arriving online

Let $t \leftarrow 2(\frac{1-\phi}{2A_m})^2 \log m \log n$;
Observe the first t agents, $T = \{\sigma(1), \ldots, \sigma(t)\}$;
For each $i = 1, \ldots, k$, let a_i be the candidate that occurs most often in position i among $\pi^{\sigma(1)}, \ldots, \pi^{\sigma(t)}$.;
Choose candidates $a_1, \ldots, a_k$ and let the process run to completion;

We now show that this algorithm does, indeed, exhibit vanishing regret as n grows large.

Theorem 12. *Suppose that $n > m^{2+\epsilon} \frac{1}{1-\phi}$ for some $\epsilon > 0$. Then algorithm $\mathcal{A}$ satisfies $E[v(\mathcal{A})] \geq (1 - o(1))OPT$.*

The proof of the theorem, which relies on the Hoeffding and the union bound, appears in the full version of the paper.

6.2 The Borda Scoring Rule

We now demonstrate that if our positional scoring function is the canonical Borda scoring function, then we can obtain a good approximation with fewer samples (and hence a weaker restriction on the size of n relative to m). In the Borda positional scoring function, for an agent with preference $\pi \in S_m$, the score is defined as follows: $B_i(a) = m - \pi(a)$; i.e. the scores are evenly spread between 0 and $m-1$.

We begin with a lemma about the Mallows model, which shows that we do not expect the top candidate to be placed very far from its position in the reference ranking:

Claim 13. *Let $\pi \sim D(\hat{\pi}, \phi)$, and let $a = \pi^{-1}(i)$; i.e. the first item in the permutation. Then with high probability $\hat{\pi}(a) = o(m)$.*

Proof. Fix $c \in (0,1)$. Now, consider the probability that any of the elements $\lfloor c \cdot m \rfloor, \ldots, m$ appear in position one in a sampled permutation π:

$$Pr[\pi(i) = 1 : i \geq \lfloor c \cdot n \rfloor] = \sum_{i=\lfloor c \cdot n \rfloor}^{m} \sum_{\pi \in S_m : \pi(i)=1} \frac{\phi^{d_K(\hat{\pi},\pi)}}{Z_m} = \sum_{i=\lfloor c \cdot n \rfloor}^{m} \frac{\phi^{i-1} \cdot Z_{m-1}}{Z_m}$$
$$= \sum_{i=\lfloor c \cdot n \rfloor}^{m} \frac{\phi^{i-1}}{1 + \phi + \cdots + \phi^{m-1}} \tag{6.2}$$

The claim follows from the fact that this is essentially a sum of exponentially small terms □

We will complement the above claim by showing that w.h.p. (albeit not necessarily exponentially small), the position of the first element in a sampled permutation in the reference ranking is bounded by $O(\log m)$. We then argue that by sampling more permutations, we can augment our bound. The claims are essentially consequences of the results obtained by Braverman and Mossel. Recall that an equivalent statement of the probability of sampling a permutation is $Pr[\pi] = e^{-\beta i}$, where $\beta = -ln\phi$.

Claim 14 ([4])**.**

$$Pr[\pi^{-1}(1) \geq i] \leq e^{-\beta i}/(1 - e^{-\beta}) \tag{6.3}$$

The proof of this claim is similar to the one of Claim 13.

Corollary 15.

$$Pr[\pi^{-1}(1) \geq \ln m] \leq m^{-\beta}/(1 - e^{-\beta}) \tag{6.4}$$

The following claim argues that the error in our estimate for the first element in $\hat{\pi}$ goes linearly small with the number of sampled permutations $\sigma^1, \ldots, \sigma^r \sim D(\hat{\pi}, \phi)$.

Claim 16 ([4])**.** *Suppose that the permutations $\pi^1, \ldots, \pi^r$ are drawn from $D(\hat{\pi}, \phi)$, and let $\overline{\pi}(a) = \frac{1}{r}\sum_{i=1}^{r} \pi^i(a)$.*

$$Pr[|\overline{\pi}(\ell) - \ell| \geq i] \leq 2 \cdot \left(\frac{(5i+1) \cdot e^{-\beta i}}{1 - e^{-\beta}}\right)^r \quad , \quad \textit{for all } i \in [m]. \tag{6.5}$$

Setting $i = \ln n$, we obtain the following corollary:

Corollary 17. *Let $\alpha > 0$. Then for sufficiently large n,*

$$Pr[|\overline{\pi}(a_\ell) - \ell| \geq \frac{\alpha + 2}{\beta \cdot r} \ln n] < n^{-\alpha} \tag{6.6}$$

Despite the above results that imply that using the top-ranked element in even a single sample should get us close to the top-ranked element in the reference ranking, we still have to argue that w.h.p., this estimate also approximates the expected top-ranked element, induced by the distribution. The following result provides an affirmative answer to this question.

Theorem 18 ([4]). *Let* $L = \max\left(6 \cdot \frac{\alpha+2}{\beta \cdot r} \log m, 6 \cdot \frac{\alpha+2+1/\beta}{\beta}\right)$. *Then except with probability* $< 2 \cdot m^{-\alpha}$, *for any maximum-likelihood* π^m *and for all* ℓ, *we have*

$$|\pi^m(a_\ell) - \hat{\pi}(a_\ell)| \le 32L \tag{6.7}$$

where $\hat{\pi}$ *is the reference ranking.*

So in total, with probability $n^{-\alpha}$, $|\overline{\pi}(a_\ell) - \pi^m(a_\ell)| \le O(1)$. Thus, we get a natural algorithm for maximizing the average Borda score for all but the first $\log n$ agents:

Theorem 19. *The algorithm that samples the first* $\log n$ *permutations and puts on the slate the element from* A *with the highest average score obtains a* $1 - O(1/n)$*-approximation of the optimal average Borda score.*

The theorem follows from the previous conclusion and by recalling that the maximum value any element can receive is $m - 1$.

6.3 The case of $k \ge 1$

Here, we show that by allowing the selection of k elements from A, the probability of maximizing the expected Borda rank, increases exponentially.

Theorem 20. *Let* $\pi^1, \ldots, \pi^{\log n}$ *be a a set of* $\log n$ *sample permutations, randomly drawn from distribution* $D(\hat{\pi}, \phi)$. *And let* $\overline{\pi}$ *be their average ranking. Then*

$$Pr[\overline{\pi}(a_i) > \log n + i : \forall i \in [k]] < n^{-O(k)} \tag{6.8}$$

Proof. Let π be a permutation over A such that for all $i \in [k]$, $\pi(a_i) \ge \log n + i$. Then consider the i'th element a in π. The number of pairwise inversions that exist in π w.r.t it are at least $\log n$, by our assumption that $\hat{\pi}(a) > \log n + i$. Then by definition of the distribution, the probability of sampling such a permutation π is at most $\frac{Z_{m-k} \prod_{i=1}^{k} \phi^{\log n}}{Z_m} \le \phi^{k \cdot \log n} = n^{-O(k)}$ □

Note that the above theorem needs to be complemented with an upper bound on the gap between the reference ranking position of and the maximum-likelihood of each candidate. However, we can easily get this by sampling $r = \log n$ permutations and applying Theorem 18, which gives a maximal $O(1)$ gap between the maximum-likelihood position and the reference rank, for any element in A, with polynomially (in n) small probability. I do believe however, that the polynomially small probability of an error could be shown to be in fact exponentially small in k (i.e. $n^{-O(k)}$).

7 Conclusions and Future Directions

We have given two methods for choosing the (approximately) best candidate in two natural and standard settings for the online choice problem at hand. As we have demonstrated, even with just a budget of 1, one could obtain very good results in the Mallows model, with a relatively small sample set. In the adversarial setting, we have shown that with a relatively small sample set (albeit not logarithmically small) one can approximate the optimal choice of a candidate with up to an $o(1)$ multiplicative error, with high probability. More importantly, we have shown that by taking a sampling approach we can approximate the social optimum, whenever the voting rule is a positional scoring function. As a result, this gives a useful tool when moving to a utilitarian setting.

One direction for future investigation would be to improve the rate at which the regret vanishes as n grows, both in the distributional setting as well as in the adversarial setting. Another direction that our study raises is the study of more involved constraints. In particular, we believe that if the alternatives have associated costs, then one could extend our work to cases in which there is a knapsack constraint. In terms of the our original example, we could imagine that there are costs attributed to the construction of the manufacturing lines. Furthermore, we can imagine that there are unit costs for producing copies of the alternatives in their production lines. More precisely, the decision maker will pay an initial price t_a for setting up the production line for alternative a, as well as an additional price of l_a for manufacturing a copy of alternative a for each agent who selects it.

The majority of our work in this paper deals with voting rules that are based on positional scoring functions, and we have shown how to extend our approaches to settings in which there are underlying utilities that induce the agent preferences. However, it will be interesting to consider settings where the voting rule is based on a non-positional scoring function.

Also, one could lift the constraint that requires the decision to be irrevocable, i.e. once an item is added, it cannot be replaced by another item. In this case, one could observe that such a model resembles the online learning setting (e.g. [17]). Alternatively, as previously studied in [1] for related settings, we can consider a setting in which the irrevocability of the decisions is relaxed. Specifically, we would like to consider the case that the decision maker is allowed to remove alternatives from the slate at a cost.

We could also extend our work by considering cases in which the agents can strategically delay their arrival, so as to increase their payoffs due to having a larger set of selected alternatives. Clearly, the pure sampling approach we have taken in this paper would be problematic, as none of the agents would like to take part in the initial sampling of preferences, and would thus delay their arrival in order to avoid it. Also, this scenario may tie-in with the previous extension, so that the agents, who can delay their arrivals, will be somewhat discouraged to do so due to a more powerful algorithm.

Acknowledgements

We would like to thank Ariel Procaccia for his many helpful comments and for pointing out the connection between this work and the results on low-distortion choice rules in [3]. We would also like to thank Craig Boutilier for many fruitful discussions.

References

[1] Moshe Babaioff, Jason D. Hartline, and Robert D. Kleinberg. Selling ad campaigns: online algorithms with cancellations. In *ACM Conference on Electronic Commerce*, pages 61–70, 2009.

[2] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. Online auctions and generalized secretary problems. *SIGecom Exchanges*, 7(2), 2008.

[3] Craig Boutilier, Ioannis Caragiannis, Simi Haber, Tyler Lu, Ariel D. Procaccia, and Or Sheffet. Optimal social choice functions: A utilitarian view. *To appear on Proc. 13th ACM Conference on Electronic Commerce, Jun 2012*, 2012.

[4] Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 268–276, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[5] John R. Chamberlin and Paul N. Courant. Representative deliberations and representative decisions: Proportional representation and the borda rule. *The American Political Science Review*, 77(3):pp. 718–733, 1983.

[6] Jean-Paul Doignon, Aleksandar Pekeč, and Michel Regenwetter. The repeated insertion model for rankings: Missing link between two subset choice models. *Psychometrika*, 69:33–54, 2004. 10.1007/BF02295838.

[7] Eyal Even-Dar, Robert Kleinberg, Shie Mannor, and Yishay Mansour. Online learning for global cost functions. In *COLT*, 2009.

[8] M. A. Fligner and J. S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):pp. 359–369, 1986.

[9] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching with unknown distributions. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 587–596, New York, NY, USA, 2011. ACM.

[10] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358, 1990.

[11] Robert D. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *SODA*, pages 630–631, 2005.

[12] Tyler Lu and Craig Boutilier. Budgeted social choice: From consensus to personalized decision making. In *IJCAI*, pages 280–286, 2011.

[13] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 597–606, New York, NY, USA, 2011. ACM.

[14] C. L. Mallows. Non-null ranking models. *Biometrika*, 44(1/2):pp. 114–130, 1957.

[15] Ariel D. Procaccia and Jeffrey S. Rosenschein. The distortion of cardinal preferences in voting. In *The Tenth International Workshop on Cooperative Information Agents (CIA)*, 2006.

[16] Moshe Tennenholtz. Transitive voting. In *Proceedings of the 5th ACM conference on Electronic commerce*, EC '04, pages 230–231, New York, NY, USA, 2004. ACM.

[17] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

Brendan Lucier
Microsoft Research, New England
Email: `brlucier@microsoft.edu`

Joel Oren
Department of Computer Science,
University of Toronto
Email: `oren@cs.toronto.edu`

Resistance to bribery when aggregating soft constraints

Alberto Maran, Maria Silvia Pini, Francesca Rossi, and K. Brent Venable

Abstract

We consider a multi-agent scenario, where the preferences of several agents are modelled via soft constraint problems and need to be aggregated to compute a single ”socially optimal” solution. We study the resistance of various ways to compute such a solution to influence the result, such as those based on the notion of bribery. In doing this, we link the cost of bribing an agent to the effort needed by the agent to make a certain solution optimal, by only changing preferences associated to parts of the solution. This leads to the definition of four notions of distance from optimality of a solution in a soft constraint problem. The notions differ on the amount of information considered when evaluating the effort.

1 Introduction

Often agents need to cooperate, rather than compete, in order to take a collective decision. By doing this, the decision can be better than what they would have chosen had they reasoned in isolation. Examples are collections of experts that submit their suggestions on what to do, which are then aggregated to obtain a single suggestion. Such experts could be, for example, classifiers in machine learning tasks, or web page rankers in web search. To make a very concrete example, when looking for a hotel in a certain city, often we use systems that exploit several different search engines, each one reporting a hotel ranking. Such rankings are usually reported to the user as they are, while it would be more useful if they were aggregated to get a collective hotel ranking from where to choose.

In this paper we study such scenarios, modelled by a collection of agents that express their preferences over a common set of solutions to a problem. We assume that such preferences are modelled by soft constraints. To consider concrete instances of soft constraints, we focus on fuzzy and weighted constraints. The agents' preferences are then aggregated to compute a single ”socially optimal” solution. To model this, we consider some voting rules. Although voting rules have been defined and studied in the context of political elections, they do exactly what we want: aggregating individual's preferences into a single collective ”winner”. We then study the resistance of this setting, considering different voting rules, to external or internal attempts to influence the result. This happens often in political elections, but it could occur also in our settings.

For example, when voting on a Doodle event to choose a date for a meeting, if one participant sees how the others have voted (and thus can compute the result by considering these votes and her true vote), she could vote in a strategic way (that is, differently to what her true vote would say) to get a better result for her. This example is an instance of the so-called manipulation, where one or more agents may misreport their votes to get a better solution. Other attempts to influence the result, usually referred to as ”control”, may come from a chair of the voting process, who can have the power, to set the number of voters, or the candidate decisions, or the voting rule to use.

A third kind of attempt may come from an external agent, usually called the ”briber”, who has a preferred solution, and tries to get that solution as the result of the voting process, by paying some agents to vote in a certain way, and by doing this while staying within its budget. In defining bribing scenarios, it is thus necessary to decide what the briber can ask an agent to do (for example, just making a certain candidate optimal, or changing more of its preference ordering) and how costly it is for the briber to submit a certain request. The cost usually represents the effort the agent has to make to satisfy the briber's request.

Classical results on voting theory tell us that every voting rule can be influenced by such attempts. However, for some voting rules, it may be computationally difficult for the manipulators, or the chair, or the briber to understand how to design the attempt. Such rules are then said to be resistant to these attempts.

In this paper we study whether our soft constraint aggregation scenarios are resistant to bribery. We consider two main approaches to aggregating the preferences: a sequential one, where agents vote on each variable at a time, and a one-step approach, where agents vote just once on entire solutions. We then define five cost schemes to compute the cost of satifying a briber's request. We find out that the one-step approach (which uses the Plurality voting rule) is not resistant to bribery: it is computationally easy for a briber to know whom to bribe and what to ask for, in order to make its preferred candudate win (if possible). On the other hand, the sequential approaches (which are based on voting rules such as Plurality, Approval, and Borda), are all resistant to bribery. This is very interesting, since the sequential approaches are also better in terms of complexity of determining the collective solution. As noted above, the cost schemes used in the bribery setting can be seen as a measure of the effort for an agent to respond to a briber's request. If the request is related to making a certain solution, say A, optimal (which means voting for it, if we use Plurality), then the cost can be considered a measure of how much the agent needs to change in its soft constraint problem in order to make A optimal. We assume that the agents want to do this by modifying just the preferences of parts of A, since otherwise also other solutions would be unnecessarily moved from their position in the preference ordering. We notice that studying resistance to bribing in constraint-based preference aggregation is interesting and useful in itself, but it has also a wider applicability within typical constraint programming tasks, such as computing the top *k* solutions and encoding solution preferences.

2 Background

Soft constraints. A soft constraint [15] involves a set of variables and associates a value from a (partially ordered) set to each instantiation of its variables. Such a value is taken from a c-semiring[1], which is defined by $\langle A, +, \times, 0, 1\rangle$, where A is the set of preference values, $+$ induces an ordering over A (where $a \leq b$ iff $a + b = b$), $\times$ is used to combine preference values, and 0 and 1 are respectively the worst and best element. A Soft Constraint Satisfaction Problem (SCSP) is a tuple $\langle V, D, C, A\rangle$ where V is a set of variables, D is the domain of the variables, C is a set of soft constraints (each one involving a subset of V), A is the set of preference values.

An instance of the SCSP framework is obtained by choosing a specific c-semiring. For instance, a classical CSP [15] is just an SCSP where the c-semiring is $S_{CSP} = \langle\{false, true\}, \vee, \wedge, false, true\rangle$. Choosing $S_{FCSP} = \langle[0, 1], max, min, 0, 1\rangle$ instead means that preferences are in [0,1] and we want to maximize the minimum preference. This is the setting of fuzzy CSPs (FCSPs) [15], that we will use in the examples of this paper. In the paper we will also consider the setting of weighted CSPs (WCSPs), where the c-semiring is $S_{WCSP} = \langle R^+, min, +, +\infty, 0\rangle$, i.e., preferences are interpreted as costs from 0 to $+\infty$, and we want to minimize the sum of the costs. We note that SCSPs generalize CSPs.

Figure 1 shows the constraint graph of an FCSP where $V = \{x, y, z\}$, $D = \{a, b\}$ and $C = \{c_x, c_y, c_z, c_{xy}, c_{yz}\}$. Each node models a variable and each arc models a binary constraint, while unary constraints define variables' domains. For example, c_y associates preference 0.4 to $y = a$ and 0.7 to $y = b$. Default constraints such as c_x and c_z will often be omitted in the following examples.

Solving an SCSP means finding some information about the ordering induced by the constraints over the set of all complete variable assignments. In the case of FCSPs and WSCSPs, such an ordering is a total order with ties. In the example above, the induced ordering has $(x = a, y = b, z = b)$ and $(x = b, y = b, z = b)$ at the top, with preference 0.5, $(x = a, y = a, z = a)$ and $(x = b, y = a, z = a)$ just below with 0.4, and all others tied at the bottom with preference 0.2. An

[1] This is just a semiring with additional properties motivated by constraint reasoning.

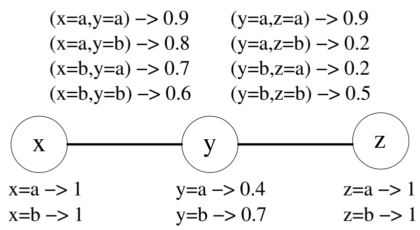


Figure 1: A tree-shaped FCSP.

optimal solution, say s, of an SCSP is then a complete assignment with an undominated preference (thus $(x = a, y = b, z = b)$ or $(x = b, y = b, z = b)$ in this example). Given a variable x, we write $s \downarrow x$ to denote the value of x in s.

Given an FCSP Q and a preference α, we will denote as $cut_\alpha(Q)$ the CSP obtained from Q allowing only tuples with preference greater than or equal to α. The set of solutions of Q with preference greater than or equal to α coincides with the set of solutions of $cut_\alpha(Q)$.

Finding an optimal solution is an NP-hard problem, unless certain restrictions are imposed, such as a tree-shaped constraint graph. Constraint propagation may help the search for an optimal solution. Given a variable ordering o, an FCSP is directional arc-consistent (DAC) if, for any two variables x and y linked by a fuzzy constraint, such that x precedes y in the ordering o, we have that, for each a in the domain of x, $f_x(a) = max_{b \in D(y)}(min(f_x(a), f_{xy}(a, b), f_y(b)))$, where f_x, f_y, and f_{xy} are the preference functions of c_x, c_y and c_{xy}. This definition can be generalized to any instance of the SCSP approach by replacing max with $+$ and min with $\times$. Therefore, for WCSPs it is sufficient to replace max with min and min with sum.

DAC is enough to find the preference level of an optimal solution when the problem has a tree-shaped constraint graph and the variable ordering is compatible with the father-child relation of the tree [15]. In fact, such an optimum preference level is the best preference level in the domain of the root variable.

Voting rules. A voting rule allows a set of voters to choose one among a set of candidates. Voters need to submit their vote, that is, their preference ordering (or part of it) over the set of candidates, and the voting rule aggregates such votes to yield a final result, usually called the winner. In the classical setting [2], given a set of candidates C, a *profile* is a collection of total orderings over the set of candidates, one for each voter. Given a profile, a *voting rule* maps it onto a single winning candidate (if necessary, ties are broken appropriately). In this paper, we will often use a terminology which is more familiar to multi-agent settings: we will sometimes call "agents" the voters, "solutions" the candidates, and "decision" or "best solution" the winning candidate.

Some examples of widely used voting rules, that we will study in this paper, are:

- *Plurality*: each voter states a single preferred candidate, and the candidate who is preferred by the largest number of voters wins;
- *Borda*: given m candidates, each voter gives a ranking of all candidates, the i^{th} ranked candidate gets a score of $m - i$, and the candidate with the greatest sum of scores wins;
- *Approval*: given m candidates, each voter approves between 1 and $m - 1$ candidates, and the candidate with most votes of approval wins.

We know that every voting rule is manipulable [2]. However, if it is computationally difficult to influence the result by using a certain voting rule, we can say that the voting rule is *resistant* to such attempts. Thus the computational complexity of various attempts to influence the result of the voting process has been studied [3, 11, 6, 14, 12]. Besides manipulation, which refers to scenarios where there is a voter (or a group of voters) who can get a better result by lying about its preference ordering, another kind of attempt to influence the result is called *bribery*: there is an outside agent, called the briber, that wants to affect the result of the election by paying some voters to change their votes, while being subject to a limitation of its budget.

Sequential preference aggregation. Assume to have a set of agents, each one expressing its preferences over a common set of objects via an SCSP whose variable assignments correspond to the objects. Since the objects are common to all agents, this means that all the SCSPs have the same set of variables and the same variable domains but they may have different soft constraints, as well as different preferences over the variable domains. In [8] this is the notion of *soft profile*, which is formally defined as a triple (V, D, P) where V is a set of variables (also called issues), D is a sequence of $|V|$ lexicographically ordered finite domains, and P a sequence of m SCSPs over variables in V with domains in D[2]. A *fuzzy profile* (resp., *weighted profile*) is a soft profile with fuzzy (resp., weighted) soft constraints. An example of a fuzzy profile where $V = \{x, y\}$, $D_x = D_y = \{a, b, c, d, e, f, g\})$, and P is a sequence of seven FCSPs, is shown in Fig. 2.

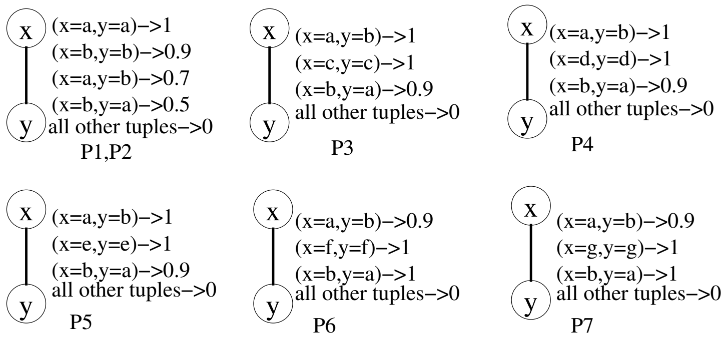


Figure 2: A fuzzy profile.

The idea proposed in [8, 7] to aggregate the preferences in a soft profile in order to compute the winning variable assignment is to sequentially vote on each variable via a voting rule, possibly using a different rule for each variable. Given a soft profile (V, D, P), assume $|V| = n$, and consider an ordering of the variables $O = \langle v_1, \ldots, v_n \rangle$, and a corresponding sequence of voting rules $R = \langle r_1, \ldots, r_n \rangle$ (that will be called "local rules"). The sequential procedure is a sequence of n steps, where at each step i,

- All agents are first asked for their preference ordering over the domain of variable v_i, yielding profile p_i over such a domain. To do this, the agents achieve DAC on their SCSP, considering the ordering O.
- Then, the voting rule r_i is applied to profile p_i, returning a winning assignment for variable v_i, say d_i. If there are ties, the first one following the given lexicographical order will be taken.
- Finally, the constraint $v_i = d_i$ is added to the preferences of each agent and DAC is achieved to propagate its effect considering the reverse ordering of O.

After all n steps have been executed, the winning assignments are collected in the tuple $\langle v_1 = d_1, \ldots, v_n = d_n \rangle$, which is declared the winner of the election. This is denoted by $Seq_{O,R}(V, D, P)$.

[2]Notice that a soft profile consists of a collection of SCSPs over the same set of variables, while a profile (as in the classical social choice setting) is a collection of total orderings over a set of candidates.

A sequential approach similar to this one has been considered in [13], where however agents' preferences are expressed via CP-nets.

In the soft profile shown in Figure 2, assume the variable ordering is $\langle x, y \rangle$ and r_i = Approval for all $i = 1, 2$. In step 1, agents achieve DAC. This changes the preferences of the agents over x. For example, in P_1 and P_2, $x = a$ maintains preference 1, $x = b$ gets preferences 0.9, and all other domain values get preference 0, while in P_3, $x = a$ and $x = c$ maintain preference 1, $x = b$ gets preference 0.9, while all other values get preference 0. Then, Approval is applied on the profile over x where the sets of approved values are all the optimals: $\{a\}$ for the first two voters and respectively $\{c, a\}$, $\{d, a\}$, $\{e, a\}$, $\{f, b\}$, and $\{g, b\}$ for the others. Thus, $x = a$ is chosen and the constraint $x = a$ is added to all SCSPs, and its effect is propagated by achieving DAC on the domain of y. In step 2, achieving DAC does not modify any preference (since y is the last variable) and the set of approved values for y is $\{a, b\}$ for P_1 and P_2 and $\{b\}$ for the other agents. Thus the elected solution with the sequential procedure is $s = (x = a, y = b)$, which has preference 0.7 for P_1 and P_2, 1 for P_3, P_4, and P_5, and 0.9 for P_6 and P_7.

An alternative to this sequential procedure would be to generate the preference orderings for each voter from their FCSPs, and then to aggregate them in one step via a voting rule, for example Approval. In our example, $(x = a, y = b)$ gets 3 votes (that is, it is optimal for 3 agents), $(x = a, y = a)$ and $(x = b, y = a)$ each gets 2 votes, $(x = f, y = f)$, $(x = d, y = d)$, $(x = c, y = c)$, $(x = e, y = e)$, and $(x = g, y = g)$ each gets 1 vote, while all other solutions get no vote. Thus the winner is $(x = a, y = b)$.

3 The bribery problem

We consider scenarios where a collection of agents need to take a decision, by selecting it out of a set of possible decisions, that are described by the Cartesian product of the domains of a set of variables. These variables are shared by all agents. Each agent has its own preferences over such decisions, described via a set of soft constraints and charges the briber for changing his preferences according to a cost scheme. In this paper, by soft constraints we mean either fuzzy or weighted constraints. Also, we assume that all agents have tree-shaped soft constraints problems. Note that the set of solutions of such constraint problems (that is, the set of decisions among which to choose one) is in general exponentially large w.r.t. the size of the soft constraint problems. We also assume that the number of such solutions is exponentially large w.r.t. the number of agents. We now define the bribery problem of which we will study the computational complexity:

Definition 1 *Given a voting rule V and a cost scheme C, we denote by (V, C)-Bribery the problem of determining if it is possible to make a preferred candidate win, when voting rule V is used, by bribing agents and by spending less than a certain budget according to cost scheme C.*

3.1 Winner determination

It makes sense to consider only winner determination approaches which are polynomial to compute: if it is difficult to compute the winning decision, it is also difficult for a briber to compute how to bribe the agents (since he needs to know who the winner is without the bribery). We consider two main approaches: sequential and one-step. For the sequential approach, we employ the sequential voting procedure described in the previous section. We have an ordering O over the variables, and we are going to consider each variable in turn in such an ordering. At each step, each agent provides some information about the considered variable, say X, which depends on the voting rule we use:

- Sequential Plurality (SP): one best value for X;
- Sequential Approval (SA): all best values for X;

- Sequential Borda (SB): a total order (possibly with ties) over the values of X, along with the preference values for each domain element.

We then choose one value for the considered variable, as follows:

- SP and SA: the value voted by the highest number of agents;
- SB: the value with best score, where the score of a value is the sum of its preferences over all the agents; notice that "best" here means maximal in the case of fuzzy constraints, while it is the minimal in the case of weighted constraints.

Once a value is chosen for a variable, this value is broadcasted to all agents, who fix variable X to this value in their soft constraints and achieve DAC in the reverse ordering w.r.t. O. We then continue with the next variable, and so on until all variables have been handled.

The alternative to a sequential approach is a one-step approach, where each agent votes over decisions regarding all variables, not just one at a time. In this case, a possible voting rule to use is what we call One-step Plurality (OP), where each agent provides an optimal solution of his soft constraint problem, and then we select the solution which is provided by the highest number of agents.

For all the voting rules we consider (SP, SA, SB, and OP), it is computationally easy for an agent to vote. An approach like OP is however less satisfactory that the sequential approaches in terms of *ballot expressiveness*: since the number of solutions is exponentially large with respect to the number of agents, there is an exponential number of solutions which are not voted by any agent. However, if we want agents to be able to compute their vote in polynomial time, we need to set a bound to the number of solutions they can vote for, and this means that in all cases an exponentially large number of solutions will not be voted. So there is trade-off between easiness of computing votes and ballot expressiveness.

We don't consider one step Approval since voting could require exponential time due to the fact that each agent may have an exponential size set of optimals.

3.2 Bribery actions and cost schemes

If we use Plurality to determine the winner, either in its sequential or one-step version, the most natural request a briber can have for an agent is to ask the agent to make a certain solution (or a certain value in the sequential case) optimal in his soft constraint problem. In order to do this, the agent can modify the preference values inside its variable domains and/or constraints.

To define the cost of a briber's request, which is to make a certain solution A optimal, we consider the following approaches:

- C_{equal}: The cost is fixed (without loss of generality, we will assume it is 1), no matter how many changes are needed to make A optimal;
- C_{do}: The cost is the distance from the preference of A, denoted with $pref(A)$, to the optimal preference of the soft constraint problem of the agent, denoted with *opt*. If we are dealing with fuzzy numbers and we may prefer to have integer costs, the cost will be defined as $C_{do} = (opt - pref(A)) * l$, where l is the number of different preference values allowed. With weighted constraints, if costs are natural numbers, we may define $C_{do} = pref(A) - opt$, since *opt* is the smallest cost.
- C_{don}: The cost is determined by considering both the distance between the preference of A and the optimal preference, and the number of parts of A, say t, that correspond to the projections of A over the constraints, that must be modified in order to make A optimal. Thus, if we have n variables, with fuzzy constraints we may define $C_{don} = ((opt - pref(A)) * l *$

$M) + t$, where M is a large integer and $1 \leq t \leq 2n - 1$. If instead we consider weighted constraints, we define $C_{don} = ((pref(A) - opt) * M) + t$. In both cases, the role of M is to assure a higher bribery cost for a less preferred candidate: we want that the highest cost at a given preference level for A, that is, $d * M + 2n - 1$, where $d = (opt - pref(A)) * l$ and n is the number of variables, to be smaller than the lowest cost at the next preference level, that is, $(d + 1)M + 1$. This yields $M > 2n - 2$.

- C_{dow}: The cost is computed by considering the same as in C_{don}, but each preference to be modified is associated with a cost proportional to the change required on that preference. If we denote by t_i any tuple of A with preference $\leq opt$, then the cost will be $((opt - pref(A)) * l * M) + \sum_{t_i}(opt - pref(t_i)) * l$ for fuzzy constraints, where the role of M is similar to the one in C_{don}. For weighted constraints, we analogously define $C_{dow} = ((pref(A) - opt) * M) + \sum_{t_i}(pref(t_i) - opt)$. However, it is easy to see that $\sum_{t_i}(pref(t_i) - opt) = pref(A) - opt$, thus we have $C_{dow} = ((pref(A) - opt) * (M + 1))$.
- C_{donw}: The cost is the combination of C_{don} and C_{dow}. For fuzzy constraints: $C_{donw} = ((opt - pref(A)) * l * M) + t * M' + \sum_{t_i}(opt - pref(t_i)) * l$, where M' has a similar role as M w.r.t. the second and third component of the sum. For weighted constraints: $C_{donw} = ((pref(A) - opt) * M) + t$ (by simplifying as in C_{dow}).

4 Winner and cost determination are both computationally easy

We are now ready to prove formally that, for all the voting rules we consider, winner determination is computationally easy. As noted earlier, if it were computationally difficult, bribery would necessarily be computationally difficult, so it would not be interesting to study the complexity of bribery. If instead winner determination is computationally easy, we may wonder if the voting rule is resistant to bribery (that is, bribery is computationally difficult) or not.

Theorem 1 *Winner determination takes polynomial time for SP, SA, SB, and OP when agents' preferences are tree-shaped fuzzy or weighted CSPs.*

Proof: For each variable, SP (resp., SA) requires most preferred value(s) in the domain of that variable. SB instead requires an ordering over such values. The fact that we are considering tree-shaped soft constraint problems ensures that voting, in all these cases, can be done in polynomial time by achieving DAC. Winner determination is then polynomial as well, since it just requires a number of polynomial steps which equals the number of variables. For OP, computing an optimal solution is polynomial on tree-shaped soft constraint problems, so voting is polynomial. Determining the winner requires just counting the number of votes for each of the voted candidates (which are in polynomial number), so it is polynomial as well. □

It is polynomial also to compute the cost to respond to a briber's request, for all our cost schemes.

Theorem 2 *Given a tree-shaped fuzzy or weighted CSP and an outcome A, determining the cost to make A an optimal outcome takes polynomial time for C_{equal}, C_{do}, C_{don}, C_{dow}, and C_{donw} .*

Proof: We can check if A is already optimal in polynomial time by first computing the optimal preference *opt* and then checking if it coincides with the preference of A, denoted $pref(A)$. If so, the cost is 0. Otherwise, with C_{equal} the cost is always 1. To compute the cost according to C_{do}, C_{don}, C_{dow}, and C_{donw}, we need to compute *opt*, the numbers of tuples of A with preference worse than *opt*, and the distance of their preferences from *opt*. All of these components can be computed in polynomial time for tree-shaped problems. □

5 Resistance to bribery when voting with SP, SA, and SB

We can now study the resistance to bribery of the voting rules we consider, that is, SP, SA, SB, and OP. Here we consider SP, SA, and SB. We recall that agents vote with tree-shaped fuzzy or weighted CSPs.

Theorem 3 *(V, C)-Bribery is NP-complete (and also W[2]-complete with parameter being the budget) for $V \in \{SP, SA, SB\}$ and $C \in \{C_{equal}, C_{do}\}$.*

Proof: Membership in NP is easy to prove. To show completeness, we provide a polynomial reduction from the OPTIMAL LOBBYING (OL) problem [5]. In this problem, we are given an $m \times n$ $0/1$ matrix E and a $0/1$ vector $\vec{x}$ of length n where each column of E represents an issue and each row of E represents a voter. We say E is a binary approval matrix with 1 corresponding to a "yes" vote and $\vec{x}$ is the target group decision. We then ask if there a choice of k rows of the matrix E such that these rows can be edited so that the majority of votes in each column matches the target vector $\vec{x}$. This problem is shown to be $W[2]$-complete with parameter k. By giving a polynomial reduction from OL to our bribery problem, we show that our problem is NP-complete (actually $W[2]$-complete with parameter being the budget B). Given an instance $(E, \vec{x}, k)$ of OL, we construct an instance of (V-C_{do})-Bribery, where $V \in \{SP, SA, SB\}$, containing constraints with only independent binary variables. The number of variables, n, is equal to the number of columns in E. For each row of E, we create a voter with the preferences over the n variables as described in the row of E. In particular, for each variable the value indicated in the row will be associated with preference 1 while the other value will be associated with preference 0. Thus, each voter has a unique most preferred solution with preference 1 and all other complete assignments have preference 0. We set the preferred outcome $A = \vec{x}$. This means that according to C_{do}, all voters not voting for A have the same cost to be bribed, which is $(opt - pref(A)) * 2 = (1 - 0) * 2 = 2$. Finally, we set the budget $B = 2k$. With C_{equal}, the cost is always 1 if A is not already voted for. We note that since we have only two values for each variable, SP, SA and SB coincide with sequential majority, thus A wins the election if and only if there is a selection of k rows of E such that $\vec{x}$ becomes the winning agenda of the OL instance. Since both fuzzy and weighted CSPs generalize CSPs, the result holds also for such classes of soft constraints. □

Theorem 4 *(V,C)-Bribery is NP-complete (and also W[2]-complete) for $V \in \{SP, SA, SB\}$ and $C \in \{C_{don}, C_{dow}, C_{donw}\}$, if $M > n * m$, where n is the number of variables and m the number of voters.*

Proof: We use a reduction similar to the one described for Thm. 3 from the optimal lobbying problem. In particular the structure of the soft profile is the same. The only things that vary are the costs for each voter and the budget. With fuzzy constraints, assume that we have l different levels of preferences and let us denote with d_i the positive integer $(opt_i - pref(A)) * l$, were i varies over the voters. For C_{don}, the cost for voter i is $d_i * M + t_i$ where t_i is the number of tuples where the candidate voted by voter i differs from A. For C_{dow}, the cost is $d_i * M + \sum_{t \in Diff_i(A)} (opt_i - pref(t))$, where $Diff_i(A)$ is the set of tuples in the soft constraint problem of agent i which not belong to A. Let us define budget B to be $B = kl(M + n)$ for fuzzy constraints and $B = k(M + n)$ for weighted constraints. Since we have only binary variables, SP, SA and SB coincide with sequential majority. There is a bribery strategy that does not exceed B if and only if there is a way to change at most k rows to solve the OL problem. We note that requiring $M > n * m$ is of key importance for the connection between the budget B and the modifications of k rows. For C_{donw}, the cost is $d_i * M + t_i * M' + \sum_{t_i \in Diff_i(A)} (opt_i - pref(t_i))$. Here a similar constraint for M' would work for the reduction. For weighted constraints, a similar reasoning works as well. □

6 Resistance to bribery when voting with OP

We now consider the one-step approach to aggregate the soft constraint problems, via voting rule OP. In the proof of our main theorem we need to compute n cheapest alternative candidates for an agent to vote for. We will thus start by studying the computational complexity of this task.

6.1 Computing the k cheapest candidates

We start by considering C_{do} and by showing that computing a set of k cheapest candidates according to this cost scheme is computationally easy. This will then be used also to compute a set of k cheapest candidates according to C_{equal}.

Theorem 5 *Given a tree-shaped fuzzy or weighted CSP, computing a set of k cheapest outcomes according to C_{do} and C_{equal} is in $\mathcal{P}$ when k is given in unary.*

Proof: The cost of an outcome according to C_{do} is an integer proportional to the distance between the preference of the outcome and the preference of an optimal outcome. In order to compute k cheapest solutions, we assume to have a linear order over the variables and the values in their domains. Such linear orders can be provided by the agent or can be chosen by the system. They do not need to be the same for all agents. For tree-shaped fuzzy CSPs, it has been shown in [4] that, given such linear orders and an outcome s, it is possible to compute, in polynomial time, the outcome following s in the induced lexicographic linearization of the preference ordering over the outcomes. The procedure that performs this is called Next. Thus, in order to compute k cheapest according to C_{do}, we compute the first optimal outcome according to the linearization and then we generate the set of k cheapest candidates by applying Next $k-1$ times (each time on the outcome of the previous step). Similarly, computing the k best solutions of a weighted CSP can be done in polynomial time by using the procedure suggested in [9]. If we consider C_{equal}, an agent will not charge the briber for changing his vote to another optimal candidate and will charge a fixed cost to change his vote in favor of any other (non-optimal) candidate. Thus any of the above procedures can be used (although, if k exceeds the cardinality of the set of optimal solutions, the remaining ones could, in principle, be generated randomly in a much faster way). □

Theorem 6 *Given a tree-shaped weighted CSP, computing a set of k cheapest outcomes according to C_{dow} is in $\mathcal{P}$ when k is given in unary.*

Proof: This result follows immediately from the fact that, for weighted CSPs, C_{dow} is proportional to C_{do}. □

We now consider the other cost schemes. We start by describing a general algorithm, which we call $KCheapest$, that will work for C_{don}, as well as for C_{dow} and for C_{donw} via small modifications. In what follows we assume that a voter represents his preferences with a tree-shaped fuzzy CSP. The input to $KCheapest$ is a tree-shaped fuzzy CSP P, an integer k, and a cost scheme C. The output is a set of k cheapest solutions of P according to C. $KCheapest$ performs the following steps:

1. **Find k optimal solutions of P, or all optimal solutions if they are less than k.** If the number of solutions found is k, we stop, otherwise let k' be the number of remaining solutions to be found.

2. **Look for the remaining top solutions within non-optimal solutions**. More in detail, until k' best solutions have been found or all solutions of P have been exhausted, consider each preference pl associated to some tuple in P in decreasing order and, for each tuple t of P with preference pl, perform the following:

   (a) Compute the new fuzzy CSP, P_t, obtained by fixing the tuple in the constraint (that is, by forbidding all other tuples in that constraint).

(b) Compute a new soft CSP, say P_t^w, associated to P_t, defined as follows:

i. the constraint topology of P_t^w and P_t coincide;

ii. each tuple with a preference greater than or equal to *opt* in P_t has weight 0 in P_t^w;

iii. each tuple with a preference pt such that $pl \leq pt < opt$ in P_t has weight c in P_t^w defined as follows: $c = 1$ if $C = C_{do}$, $c = pt - opt$ if $C = C_{dow}$ and $c = (1, pt - opt)$ if $C = C_{donw}$;

iv. each tuple with preference less than *pl* in P_t has weight $+\infty$ in P_t^w.

Thus, P_t^w is a weighted CSP if $C = C_{don}$ or $C = C_{dow}$, while it is a SCSP defined on the Cartesian product of two weighted semirings if $C = C_{donw}$.

(c) Compute the k' best solutions of all the solutions if they are less than k' of P_t^w.

Take the k' top solutions (or all solutions if less than k') among the sets of best solutions computed for P_t^w, $\forall t$ such that $pref(t) = pl$.

Theorem 7 *Given a tree-shaped fuzzy CSP P, computing a set of k cheapest outcomes according to C_{don}, C_{dow}, and C_{donw} is in $\mathcal{P}$ when k is given in unary.*

The above statement can be proven by showing that the solutions returned by algorithm $KCheapest$ are indeed the k cheapest (or all the solutions if the k exceeds the total number of solutions) according to the selected cost scheme (depending on how the weights are defined in step (iii)) and that $KCheapest$ runs in polynomial time.

6.2 Bribery with OP is easy

Faliszewski [10] shows that bribery when voting with plurality in single variable elections with non-uniform cost schemes is in $\mathcal{P}$ through the use of flow networks. The algorithm requires the enumeration of all candidates as part of the construction of the flow network. In our model, the number of candidates can be exponential in the size of the input, so we cannot use that construction directly. However, we show that a similar technique works by considering only a polynomial number of candidates.

Theorem 8 *(OP,C)-Bribery is in $\mathcal{P}$ for $C \in \{C_{equal}, C_{do}, C_{don}, C_{dow}, C_{donw}\}$ when agents vote with tree-shaped fuzzy CSPs and for $C \in \{C_{equal}, C_{do}, C_{dow}\}$ when agents vote with tree-shaped weighted CSPs.*

Proof: We consider all $r \in \{1, \ldots, n\}$ and ask if the bribers' favorite candidate A can be made a winner with exactly r votes without exceeding its budget B. If there is at least one r such that this is possible, then it means that the answer to the bribery problem is yes, otherwise it is no. We show that, for each r, the corresponding decision problem can be solved in polynomial time. This means that the overall bribery problem is in $\mathcal{P}$. To solve the decision problem for a certain r, we transform this problem to a minimum-cost flow problem [1]. The network has a source s, a sink t, and three "layers" of nodes.

The first layer has one node for each voter $v_1, \ldots, v_n$. There are also n edges (s, v_i), with capacity 1 and cost 0.

For the second layer of nodes, for each voter in the given profile, we add in this second layer nodes corresponding to A, to all the candidates with at least one vote (at most n), and to the n non-voted cheapest candidates for this voter, according to the cost scheme, thus adding at most $2n + 1$ candidates for each voter. Intuitively, this second layer models the profile modified by the bribery, where each voter can change its vote or also maintain the previous one. The important point is that the non-voted candidates that we do not include in the second layer can be avoided since not using them does not increase the cost of the bribery. Providing n non-voted candidates for each voter is enough, since there are n voters and in the worst case each of them has to vote for a different candidate. For each node S_{ij} in the second layer corresponding to voter v_i, we add an edge from v_i to S_{ij} with capacity $+\infty$ and cost equal to the cost of bribing v_i to vote for the candidate corresponding to node S_{ij}. Finding such candidates, and the cost for the voter to vote for them, takes polynomial time, no matter the cost scheme. Finding the voted candidates is easy since finding the optimal outcome in tree-shaped fuzzy or weighted CSPs takes polynomial time. Finding the n cheapest non-voted candidates, can be done by applying the procedures described in Section 6.1. In general, it is sufficient to compute the $2n$ cheapest candidates in order to make sure we have at least n non-voted candidates. Moreover, given a voter, computing the cost for such a voter to vote for one of the candidates is easy for both voted and non-voted candidates given the results in Section 4.

In the third layer of the network, we add a node for each candidate who already appears somewhere in the network (up to $n^2 + n + 1$). One of these nodes represents A. These third layer nodes are the nodes that enforce the constraint that no candidate besides A can receive more than r votes. These nodes have an edge from the nodes of the second layer representing the same candidate, with zero cost and infinite capacity. The output link from each of the third layer nodes to the sink has capacity r. The cost is 0 for the edge from A to the sink, while for all other candidates it is a large integer M to force as much flow through the node A as possible.

If we had included nodes for all the candidates in the second layer, we would have used a network equivalent to the one used in the proof of Theorem 3.1 in [10], which shows that there is a minimum cost flow of value n if and only if there is a way to solve the bribery problem. However, since we have a number of candidates which is superpolynomial in the size of the input, we would not have a polynomial algorithm. By including only the cheapest n alternative candidates for each voter, along with A and all the voted candidates, the result still holds. In fact, assume there is a minimum-cost flow in the larger network which goes through one of the nodes which we omit. This means that a voter has been forced to vote for another, more expensive, non-voted candidate since all its cheapest candidates had already r votes each. However, this is not possible, since we have only a total of $n - 1$ votes that can be given by the other voters, and we provide n non-voted candidates. We will build, at worst, n networks with $O(n^2)$ nodes and $O(n^3)$ edges. Since minimum-cost feasible flow problem can be solved in polynomial time in the number of nodes and edges using for example the Edmonds-Karp algorithm [1], the overall running time of this method is polynomial. □

7 Conclusions

Our results about the resistance to bribery of our ways to aggregate the preferences of a collection of agents, when they are modelled via soft constraints, can be seen in Table 1. We can see that OP is not resistant to bribery, since it is computationally easy for the briber to compute who to bribe and what to ask for, and to check whether he can do it within its budget. On the other hand, the sequential approaches (SP, SA, and SB) are all resistant to bribery, if agents compute costs according to C_{equal}, C_{do}, C_{don}, C_{dow} or C_{donw} . Thus, it is clear that sequential approaches should be preferred if resistance to bribery is an important feature. Notice that, when a problem is polynomial for soft constraints, it is also so for CSPs. Thus, OP is easy to bribe also when agents use CSPs.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] K. J. Arrow and A. K. Sen amd K. Suzumura. *Handbook of Social Choice and Welfare.* North-Holland, 2002.

[3] J.J. Bartholdi, C.A. Tovey, and M.A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

	SP	SA	SB	OP
C_{equal}	NP-c	NP-c	NP-c	P
C_{do}	NP-c	NP-c	NP-c	P
C_{don}	NP-c*	NP-c*	NP-c*	P/?
C_{dow}	NP-c*	NP-c*	NP-c*	P
C_{donw}	NP-c*	NP-c*	NP-c*	P/?

Table 1: Our results. NP-c* stands for NP-complete with the restriction on M (and M' if present). When the complexity results for fuzzy constraints and weighted constraints are different, we write X/Y, where X is the complexity for fuzzy and Y is the complexity for weighted constraints.

[4] R. I. Brafman, F. Rossi, D. Salvagnin, K. B. Venable, and T. Walsh. Finding the next solution in constraint- and preference-based knowledge representation formalisms. In *Proc. KR 2010*, 2010.

[5] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.

[6] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *JACM*, 54(3):1–33, 2007.

[7] G. Dalla Pozza, M. S. Pini, F. Rossi, and K. B. Venable. Multi-agent soft constraint aggregation via sequential voting. In *IJCAI*, pages 172–177, 2011.

[8] G. Dalla Pozza, F. Rossi, and K. B. Venable. Multi-agent soft constraint aggregation:a sequential approach. In *ICAART*, pages 277–282, 2011.

[9] N. Flerova E. Rollon and R. Dechter. Inference schemes for m best solutions for soft CSPs. In *Proc. SOFT'11*, 2011.

[10] P. Faliszewski. Nonuniform bribery. In *Proc. AAMAS 2008*, pages 1569–1572, 2008.

[11] P. Faliszewski, E. Hemaspaandra, and L.A. Hemaspaandra. How hard is bribery in elections? *JAIR*, 35:485–532, 2009.

[12] J. Lang, M. S. Pini, F. Rossi, D. Salvagnin, K. B. Venable, and T. Walsh. Winner determination in voting trees with incomplete preferences and weighted votes. *Autonomous Agents and Multi-Agent Systems*, 25(1):130–157, 2012.

[13] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57(3):304–324, 2009.

[14] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation: Complexity results. *Artif. Intell.*, 175(7-8):1272–1289, 2011.

[15] F. Rossi, P. Van Beek, and T. Walsh. *Handbook of Constraint Programming*. Elsevier, 2006.

Alberto Maran, Francesca Rossi, K. Brent Venable
Department of Mathematics
University of Padova
Via Trieste 63, 35121, Padova, Italy
Email: amaran@studenti.math.unipd.it,\{frossi,kvenable\}@math.unipd.it

Maria Silvia Pini
Department of Information Engineering
University of Padova
Via Gradenigo 6/B, 35131 Padova, Italy
Email: pini@dei.unipd.it

An Empirical Study of Voting Rules and Manipulation with Large Datasets

Nicholas Mattei and James Forshee and Judy Goldsmith

Abstract

The study of voting systems often takes place in the theoretical domain due to a lack of large samples of sincere, strictly ordered voting data. We derive several million elections (more than all the existing studies combined) from a publicly available data, the Netflix Prize dataset. The Netflix data is derived from millions of Netflix users, who have an incentive to report sincere preferences, unlike random survey takers. We evaluate each of these elections under the Plurality, Borda, k-Approval, and Repeated Alternative Vote (RAV) voting rules. We examine the Condorcet Efficiency of each of the rules and the probability of occurrence of Condorcet's Paradox. We compare our votes to existing theories of domain restriction (e.g., single-peakedness) and statistical models used to generate election data for testing (e.g., Impartial Culture). Additionally, we examine the relationship between coalition size and vote deficit for manipulations of elections under the Borda rule. We find a high consensus among the different voting rules; almost no instances of Condorcet's Paradox; almost no support for restricted preference profiles, very little support for many of the statistical models currently used to generate election data for testing, and very small coalitions needed to promote second-place candidates to the winning position in elections.

1 Introduction

One of the most common methods of preference aggregation and group decision making in human systems is voting. Many scholars wish to empirically study how often and under what conditions individual voting rules fall victim to various voting irregularities [6, 9]. Due to a lack of large, accurate datasets, many computer scientists and political scientists are turning towards statistical distributions to generate election scenarios in order to verify and test voting rules and other decision procedures [22, 25]. These statistical models may or may not be grounded in reality and it is an open problem in both the political science and social choice fields as to what, exactly, election data looks like [24]. As the computational social choice community continues to grow there is increasing attention on empirical results (see, e.g., [25]) and we hope to address this problem with our study.

A fundamental problem in research into properties of voting rules is the lack of large data sets to run empirical experiments [20,24]. There have been studies of several distinct datasets but these are limited in both number of elections analyzed [6] and size of individual elections within the datasets analyzed [9, 24]. While there is little agreement about the frequency with which different voting paradoxes occur or the consensus between voting methods, all the studies so far have found little evidence of *Condorcet's Voting Paradox* [10] (a cyclical majority ordering) or *preference domain restrictions* such as *single peakedness* [4] (where one candidate out of a set of three is never ranked last). Additionally, most of the studies find a strong consensus between most voting rules except Plurality [6,9,20].

We begin in Section 2 with a survey of the datasets that are commonly used in the literature. We then detail in Section 3 our new dataset, including summary statistics and a basic overview of the data. We then move into Section 4 which is broken into multiple subsections where we attempt to answer many questions about voting. Section 4.1 details an analysis that attempts to answer the questions "How often does Concert's Paradox occur?", "How often does any voting cycle occur?", and a look at the prevalence of single peaked preferences and other domain restricted election profiles [4, 23]. Section 4.2 investigates the consensus between multiple voting rules. We evaluate our millions of elections under the voting rules: Plurality, Copeland, Borda, Repeated

Alternative Vote, and k-Approval. In Section 4.3 we evaluate our new dataset against many of the statistical models that are in use in the ComSoc and social choice communities to generate synthetic election data. Section 5 details an experiment we preform to investigate, empirically, the relationship between necessary coalition size and vote deficit for manipulations of the Borda rule. This paper reports on an expanded analysis in terms of number of tests and amount of data used from the previously published work by Mattei [13, 14].

2 Survey of Existing Datasets

The literature on the empirical analysis of large voting datasets is somewhat sparse, and many studies use the same datasets [9, 24]. These problems can be attributed to the lack of large amounts of data from real elections [20]. Chamberlin et al. [6] provided empirical analysis of five elections of the American Psychological Association (APA). These elections range in size from 11,000 to 15,000 ballots (some of the largest elections studied). Within these elections there are no cyclical majority orderings and, of the six voting rules under study, only Plurality fails to coincide with the others on a regular basis. Similarly, Regenwetter et al. analyzed APA data from later years [21] and observed the same phenomena: a high degree of stability among elections rules. Felsenthal et al. [9] analyzed a dataset of 36 unique voting instances from unions and other professional organizations in Europe. Recently, data from a series of elections in Ireland have been studied in a variety of contexts in social choice [12]. Under a variety of voting rules Felsenthal et al. also found a high degree of consensus between voting rules (with the notable exception of Plurality).

All of the empirical studies surveyed [6, 9, 16, 20, 21, 24] came to a similar conclusion: there is scant evidence for occurrences of Condorcet's Paradox [17]. Many of these studies find no occurrence of majority cycles (and those that find cycles find them in rates of much less than 1% of elections). Additionally, each of these (with the exception of Niemi and his study of university elections, which he observes is a highly homogeneous population [16]) find almost no occurrences of either single-peaked preferences [4] or the more general value-restricted preferences [23].

Given this lack of data and the somewhat surprising results regarding voting irregularities, some authors have taken a more statistical approach. Over the years multiple statistical models have been proposed to generate election pseudo-data to analyze (e.g., [20, 24]). Gehrlein [10] provides an analysis of the probability of occurrence of Condorcet's Paradox in a variety of election cultures. Gehrlein exactly quantifies these probabilities and concludes that Condorcet's Paradox probably will only occur with very small electorates. Gehrlein states that some of the statistical cultures used to generate election pseudo-data, specifically the Impartial Culture, may actually represent a worst-case scenario when analyzing voting rules for single-peaked preferences and the likelihood of observing Condorcet's Paradox [10]

Tideman and Plassmann have undertaken the task of verifying the statistical cultures used to generate pseudo-election data [24]. Using one of the largest datasets available, Tideman and Plassmann find little evidence supporting the models currently in use to generate election data. Additionally, Tideman and Plassmann propose several novel statistical models which better fit their empirical data.

3 The New Data

We have mined strict preference orders from the Netflix Prize Dataset [2]. The Netflix dataset offers a vast amount of preference data; compiled and publicly released by Netflix for its Netflix Prize [2]. There are 100,480,507 distinct ratings in the database. These ratings cover a total of 17,770 movies and 480,189 distinct users. Each user provides a numerical ranking between 1 and 5 (inclusive) of some subset of the movies. While all movies have at least one ranking, it is not the case that all users have rated all movies. The dataset contains every movie rating received by Netflix, from its users, between when Netflix started tracking the data (early 2002) up to when the competition was announced (late 2005). This data has been perturbed to protect privacy and is conveniently coded for use by researchers.

The Netflix data is rare in preference studies: it is more sincere than most other preference data sets. Since users of the Netflix service will receive better recommendations from Netflix if they respond truthfully to the rating prompt, there is an incentive for each user to express sincere preference. This is in contrast to many other datasets which are compiled through surveys or other methods where the individuals questioned about their preferences have no stake in providing truthful responses.

We define an election as $E(m,n)$, where m is a set of candidates, $\{c_1,\ldots,c_m\}$, and n is a set of votes. A vote is a strict preference ordering over all the candidates $c_1 > c_2 > \cdots > c_m$. For convenience and ease of exposition we will often speak in the terms of a three candidate election and label the candidates as A,B,C and preference profiles as $A > B > C$. All results and discussion can be extended to the case of more than three candidates. A voting rule takes, as input, a set of candidates and a set of votes and returns a set of winners which may be empty or contain one or more candidates. In our discussion, elections return a complete ordering over all the candidates in the election with no ties between candidates (after a tiebreaking rule has been applied). The candidates in our data set correspond to movies from the Netflix dataset and the votes correspond to strict preference orderings over these movies. We break ties according to the lowest numbered movie identifier in the Netflix set; these are random, sequential numbers assigned to every movie.

We construct vote instances from this dataset by looking at combinations of three movies. If we find a user with a strict preference ordering over the three moves, we tally that as a vote. For example, given movies A,B, and C: if a user rates movie $A = 1$, $B = 3$, and $C = 5$, then the user has a strict preference profile over the three movies we are considering and hence a vote. If we can find 350 or more votes for a particular movie triple then we regard that movie triple as an election and we record it. We use 350 as a cutoff for an election as it is the number of votes used by Tideman and Plassmann [24] in their study of voting data. While this is a somewhat arbitrary cutoff, Tideman and Plassmann claim it is a sufficient number to eliminate random noise in the elections [24]. We use the 350 number so that our results are directly comparable to the results reported by Tideman and Plassmann.

The dataset is too large to use completely ($\binom{17770}{3} \approx 1 \times 10^{12}$) so we have subdivided it. We have divided the movies into 10 independent (non-overlapping with respect to movies), randomly drawn samples of 1777 movies. This completely partitions the set of movies. For each sample we search all the $\binom{17770}{3} \approx 9.33 \times 10^{8}$ possible elections for those with more than 350 votes. For 3 candidate elections, this search generated 14,003,522 distinct movie triples in total over all the subdivisions. Not all users have rated all movies so the actual number of elections for each set is not consistent. The maximum election size found in the dataset is 24,670 votes; metrics of central tendency are presented in Tables 1 and 2.

	Set 1	Set 2	Set 3	Set 4	Set 5
Median	610.0	592.0	597.0	583.0	581.0
Mean	964.8	880.6	893.3	843.3	829.9
Max.	18,270.0	19,480.0	19,040.0	17,930.0	12,630.0
Elements	1,453,012.0	1,640,584.0	1,737,858.0	1,495,316.0	1,388,892.0

	Set 6	Set 7	Set 8	Set 9	Set 10
Median	584.0	585.0	580.0	600.0	573.0
Mean	853.2	868.4	841.3	862.7	779.2
Max.	20,250.0	24,670.0	21,260.0	17,750.0	13,230.0
Elements	1,344,775.0	931,403	1,251,478	1,500,040	1,260,164

Table 1: Summary statistics for 3 candidate elections.

Using the notion of item-item extension [11], we attempted to extend every triple found in the initial search. Item-item extension allows us to trim our search space by only searching for 4 movie combinations which contain a combination of 3 movies that was a valid voting instance. For each set we only searched for extensions within the same draw of 1777 movies, making sure to remove any duplicate extensions. The results of this search are summarized in Table 2. For 4 candidate elections, this search generated 11,362,358 distinct movie triples over all subdivisions. Our constructed datasets contains more than 5 orders of magnitude more distinct elections than all the previous studies *combined* and the largest single election contains slightly more votes than the largest previously studied election from data.

	Set 1	Set 2	Set 3	Set 4	Set 5
Median	471.0	450.0	458.0	446.0	440.0
Mean	555.6	512.2	532.7	508.0	490.2
Max.	3,519.0	2,965.0	4,032.0	2,975.0	2,192.0
Elements	1,881,695.0	1,489,814.0	1,753,990	1,122,227.0	1,032,874

	Set 6	Set 7	Set 8	Set 9	Set 10
Median	449.0	454.0	447.0	432.0	424.0
Mean	512.2	521.3	513.0	475.8	468.2
Max.	3,400.0	3,511.0	3,874.0	2,574.0	2,143.0
Elements	1,082,377.0	642,537	811,130	1,117,798	427,916

Table 2: Summary statistics for 4 candidate elections.

The data mining and experiments were performed on a pair of dedicated machines with dual-core Athlon 64x2 5000+ processors and 4 gigabytes of RAM. All the programs for searching the dataset and performing the experiments were written in C++. All of the statistical analysis was performed in R using RStudio. The initial search of three movie combinations took approximately 36 hours (parallelized over the two cores) for each of the ten independently drawn sets. The four movie extension searches took approximately 250 hours per set.

4 Analysis and Discussion

We have found a large correlation between each pair of voting rules under study with the exception of Plurality (when $m = 3,4$) and 2-Approval (when $m = 3$). A *Condorcet Winner* is a candidate who is preferred by a majority of the voters to each of the other candidates in an election [9]. The voting rules under study, with the exception of Copeland, are not *Condorcet Consistent*: they do not necessarily select a Condorcet Winner if one exists [17]. Therefore, we also analyze the voting rules in terms of their *Condorcet Efficiency*, the rate at which the rule selects a Condorcet Winner if one exists [15]. In Section 4.2 we see that the voting rules exhibit a high degree of Condorcet Efficiency in our dataset. The results in Section 4.1 show extremely small evidence for cases of single peaked preferences and very low rates of occurrence of preference cycles. Finally, the experiments in Section 4.3 indicate that several statistical models currently in use for testing new voting rules [22] do not reflect the reality of our dataset. All of these results are in keeping with the analysis of other, distinct, datasets [6,9,16,20,21,24] and provide support for their conclusions.

4.1 Preference Cycles and Domain Restrictions

Condorcet's Paradox of Voting is the observation that rational group preferences can be aggregated, through a voting rule, into an irrational total preference [17]. It is an important theoretical and practical concern to evaluate how often the scenario arises in empirical data. In addition to analyzing instances of *total cycles* (Condorcet's Paradox) involving all candidates in an election, we check for two other types of cyclic preferences. We also search our results for both *partial cycles*, a cyclic ordering that does not include the top candidate (Condorcet Winner), and *partial top cycles*, a cycle that includes the top candidate but excludes one or more other candidates [9].

Table 3 summarize the rates of occurrence of the different types of voting cycles found in 4 candidate set (3 candidate table is omitted for space). The cycle counts for $m = 3$ are all equivalent due to the fact that there is only one type of possible cycle when $m = 3$. There is an extremely low instance of total cycles for all our data ($< 0.11\%$ of all elections). This corresponds to findings in the empirical literature that support the conclusion that Condorcet's Paradox has a low incidence of occurrence. Likewise, cycles of any type occur in rates $< 0.4\%$ and therefore seem of little practical importance in our dataset as well. Our results for cycles that do not include the winner mirror the results of Felsenthal et al. [9]: many cycles occur in the lower ranks of voters' preference orders in the election due to the voters' inability to distinguish between, or indifference towards, candidates the voter has a low ranking for or considers irrelevant.

	Set 1	Set 2	Set 3	Set 4	Set 5
Partial Cycle	4,088 (0.22%)	4,360 (0.29%)	3,879 (0.22%)	1,599 (0.14%)	1,316 (0.13%)
Partial Top	2,847 (0.15%)	3,042 (0.20%)	2,951 (0.17%)	1,165 (0.10%)	974 (0.09%)
Total	892 (0.05%)	1,110 (0.07%)	937 (0.05%)	427 (0.04%)	293 (0.03%)

	Set 6	Set 7	Set 8	Set 9	Set 10
Partial Cycle	1,597 (0.15%)	1,472 (0.23%)	1,407 (0.17%)	1,274 (0.11%)	1,646 (0.38%)
Partial Top	1,189 (0.11%)	1,222 (0.19%)	1,018 (0.13%)	870 (0.08%)	1,123 (0.26%)
Total	325 (0.03%)	438 (0.07%)	331 (0.04%)	198 (0.02%)	451 (0.11%)

Table 3: Number of elections demonstrating various types of voting cycles for 4 candidate elections.

Black first introduced the notion of single-peaked preferences [4], a domain restriction that states that the candidates can be ordered along one axis of preference and there is a single peak to the graph of all votes by all voters if the candidates are ordered along this axis. Informally, the idea is that every member of the society has an (not necessarily identical) ideal point along a single axis and that, the farther an alternative is from the bliss point, the lower that candidate will be ranked. A typical example is that everyone has a preference for the volume of music in a room, the farther away (either louder or softer) the music is set, the less preferred that volume is.

This is expressed in an election as the scenario when some candidate, in a three candidate election, is never ranked last. The notion of restricted preference profiles was extended by Sen [23] to include the idea of candidates who are never ranked first (single-bottom) and candidates who are always ranked in the middle (single-mid). Domain restrictions can be expanded to the case where elections contain more than three candidates [1]. Preference restrictions have important theoretical applications and are widely studied in the area of election manipulation. Many election rules become easy to affect through bribery or manipulation when electorates preferences are single-peaked [5].

Table 4 summarizes our results for the analysis of different restricted preference profiles when $m = 3$. There is (nearly) a complete lack (10 total instances over all sets) of preference profile restrictions when $m = 4$ and near lack ($< 0.05\%$) when $m = 3$. It is important to remember that the underlying objects in this dataset are movies, and individuals, most likely, evaluate movies for many different reasons. Therefore, as the results of our analysis confirm, there are very few items that users rate with respect to a single dimension.

4.2 Voting Rules

We analyze our dataset under the voting rules Plurality, Borda, 2-Approval, and Repeated Alternative Vote (RAV). We assume the reader is familiar with the normal voting rules discussed here. We

	Set 1	Set 2	Set 3	Set 4	Set 5
Single Peaked	29 (0.002%)	92 (0.006%)	624 (0.036%)	54 (0.004%)	11 (0.001%)
Single Mid	0 (0.000%)	0 (0.000%)	0 (0.000%)	0 (0.000%)	0 (0.000%)
Single Bottom	44 (0.003%)	215 (0.013%)	412 (0.024%)	176 (0.012%)	24 (0.002%)
	Set 6	Set 7	Set 8	Set 9	Set 10
Single Peaked	162 (0.012%)	148 (0.016%)	122 (0.010%)	168 (0.011%)	43 (0.003%)
Single Mid	0 (0.000%)	0 (0.000%)	0 (0.000%)	0 (0.000%)	0 (0.000%)
Single Bottom	590 (0.044%)	147 (0.016%)	152 (0.012%)	434 (0.029%)	189 (0.015%)

Table 4: Number of 3 candidate elections demonstrating preference profile restrictions.

note that RAV is an extension of the alternative vote (AV) where the process is repeated (removing the winning candidate at each step) to generate a total order over all the candidates. A more complete treatment of voting rules and their properties can be found in Nurmi [17] or Arrow, Sen, and Suzumura [1].

We follow the analysis outlined by Felsenthal et al. [9]. We establish the Copeland order as "ground truth" in each election; Copeland always selects the Condorcet Winner if one exists and many feel the ordering generated by the Copeland rule is the "most fair" when no Condorcet Winner exists [9, 17]. After determining the results of each election, for each voting rule, we compare the order produced by each rule to the Copeland order and compute the Spearman's Rank Order Correlation Coefficient (Spearman's ρ) to measure similarity [9].

We have omitted the tables of our results for space considerations, see Mattei [13, 14] for additional details and results. For the elections with $m = 3$ and $m = 4$ we have Borda and RAV agreeing with Copeland $\approx 98\%$ of the time, on average. For Plurality, when $m = 3$ we have $\approx 92\%$ agreement with Copeland. This correlation drops to $\approx 87\%$ when we move to $m = 4$. Plurality performs the worst as compared to Copeland across all the datasets. 2-Approval does fairly poorly when $m = 3$ ($\approx 90\%$) but does surprisingly well ($\approx 96\%$) when $m = 4$. We suspect this discrepancy is due to the fact that when $m = 3$, individual voters are able to select a full $2/3$ of the available candidates. All sets had a median value of 1.0 and small standard error 0.2 for plurality and much less for all rules. Our analysis supports other empirical studies in the field that find a high consensus between the various voting rules [6, 9, 21].

There are many considerations one must make when selecting a voting rule for use within a given system. Merrill suggests that one of the most powerful metrics is Condorcet Efficiency [15]. We eliminated all elections that did not have a Condorcet Winner in this analysis. All voting rules select the Condorcet Winner a surprising majority of the time. For plurality, Borda, and RAV we have a Condorcet Efficient of $\approx 95\%$, on average. The worst case is 2-Approval, when $m = 3$, as it results in the lowest Condorcet Efficiency in our dataset ($\approx 88\%$). The high rate of elections that have a Condorcet Winner ($> 80\%$) could be an artifact of how we select elections. By virtue of enforcing strict orders we are causing a selection bias in our set: we are only checking elections where many voters have a preference between any two items in the dataset.

Overall, we find a consensus between the various voting rules in our tests. This supports the findings of other empirical studies in the field [6, 9, 21]. Merrill finds much lower rates for Condorcet Efficiency than we do in our study [15]. However, Merrill uses statistical models to generate elections rather than empirical data to compute his numbers and this is likely the cause of the discrepancy [10].

4.3 Statistical Models of Elections

We evaluate our dataset to see how it matches up to different probability distributions found in the literature. We briefly detail several probability distributions (or "cultures") here that we test.

Tideman and Plassmann provide a more complete discussion of the variety of statistical cultures in the literature [24]. There are other election generating cultures, such as weighted Independent Anonymous Culture, which generate preference profiles that are skewed towards single-peakedness or single-bottomness. As we have found no support in our analysis for restricted preference profiles we do not analyze these cultures (a further discussion and additional election generating statistical models can be found in [24]).

We follow the general outline in Tideman and Plassmann to guide us in this study [24]. For ease of discussion we divide the models into two groups: probability models (IC, DC, UC, UUP) and generative models (IAC, Urn, IAC-Fit). Probability models define a probability vector over each of the $m!$ possible strict preference rankings. We note these probabilities as $pr(ABC)$, which is the probability of observing a vote $A > B > C$ for each of the possible orderings. In order to compare how the statistical models describe the empirical data, we compute the mean Euclidean distance between the empirical probability distribution and the one predicted by the model.

Impartial Culture (IC): An even distribution over every vote exists. That is, for the $m!$ possible votes, each vote has probability $1/m!$ (a uniform distribution).

Dual Culture (DC): The dual culture assumes that the probability of opposite preference orders is equal. So, $pr(ABC) = pr(CBA)$, $pr(ACB) = pr(BCA)$ etc. This culture is based on the idea that some groups are polarized over certain issues.

Uniform Culture (UC): The uniform culture assumes that the probability of distinct pairs of lexicographically neighboring orders (that share the same top candidate) are equal. For example, $pr(ABC) = pr(ACB)$ and $pr(BAC) = pr(BCA)$ but not $pr(ACB) = pr(CAB)$ (as, for three candidates, we pair them by the same winner). This culture corresponds to situations where voters have strong preferences over the top candidates but may be indifferent over candidates lower in the list.

Unequal Unique Probabilities (UUP): The unequal unique probabilities culture defines the voting probabilities as the maximum likelihood estimator over the entire dataset. We determine, for each of the data sets, the UUP distribution as described below.

For DC and UC each election generates its own statistical model according to the definition of the given culture. For UUP we need to calibrate the parameters over the entire dataset. We follow the method described in Tideman and Plassmann [24]: first re-label each empirical election in the dataset such that the order with the most votes becomes the labeling for all the other votes. This requires reshuffling the vector so that the most likely vote is always $A > B > C$. Then, over all the reordered vectors, we maximize the log-likelihood of

$$f(N_1,\ldots,N_6;N,p_1,\ldots,p_6) = \frac{N!}{\prod_{r=1}^{6} N_r!}\prod_{r=1}^{6} p_r^{N_r} \tag{1}$$

where $N_1,\ldots,N_6$ is the number of votes received by a vote vector and $p_1,\ldots,p_6$ are the probabilities of observing a particular order over all votes (we expand this equation to 24 vectors for the $m = 4$ case). To compute the error between the culture's distribution and the empirical observations, we re-label the culture distribution so that preference order with the most votes in the empirical distribution matches the culture distribution and compute the error as the mean Euclidean distance between the discrete probability distributions.

Urn Model: The Polya Eggenberger urn model is a method designed to introduce some correlation between votes and does not assume a complete uniform random distribution [3]. We use a setup as described by Walsh [25]; we start with a jar containing one of each possible vote. We draw a vote at random and place it back into the jar with $a \in \mathbb{Z}_+$ additional votes of the same kind. We repeat this procedure until we have created a sufficient number of votes.

Impartial Anonymous Culture (IAC): Every distribution over orders has an equal likelihood. For each generated election we first randomly draw a distribution over all the $m!$ possible voting vectors and then use this model to generate votes in an election.

IAC-Fit: For this model we first determine the vote vector that maximizes the log-likelihood of Equation 1 without the reordering described for UUP. Using the probability vector obtained for

$m = 3$ and $m = 4$ we randomly generate elections. This method generates a probability distribution or culture that represents our entire dataset.

For the generative models we must generate data in order to compare them to the culture distributions. To do this we average the total elections found for $m = 3$ and $m = 4$ and generate 1,400,352 and 1,132,636 elections, respectively. We then draw the individual election sizes randomly from the distribution represented in our dataset. After we generate these random elections we compare them to the probability distributions predicted by the various cultures.

	IC	DC	UC	UUP
Set 1	0.3064 (0.0137)	0.2742 (0.0113)	0.1652 (0.0087)	0.2817 (0.0307)
Set 2	0.3106 (0.0145)	0.2769 (0.0117)	0.1661 (0.0089)	0.2818 (0.0311)
Set 3	0.3005 (0.0157)	0.2675 (0.0130)	0.1639 (0.0091)	0.2860 (0.0307)
Set 4	0.3176 (0.0143)	0.2847 (0.0113)	0.1758 (0.0100)	0.2833 (0.0332)
Set 5	0.2974 (0.0125)	0.2677 (0.0104)	0.1610 (0.0082)	0.2774 (0.0300)
Set 6	0.3425 (0.0188)	0.3027 (0.0143)	0.1734 (0.0108)	0.3113 (0.0399)
Set 7	0.3043 (0.0154)	0.2704 (0.0125)	0.1660 (0.0095)	0.2665 (0.0289)
Set 8	0.3154 (0.0141)	0.2816 (0.0114)	0.1712 (0.0091)	0.2764 (0.0318)
Set 9	0.3248 (0.0171)	0.2906 (0.0130)	0.1686 (0.0100)	0.3005 (0.0377)
Set 10	0.2934 (0.0144)	0.2602 (0.0121)	0.1583 (0.0087)	0.2634 (0.0253)
Urn	0.6228 (0.0249)	0.4745 (0.0225)	0.4745 (0.0225)	0.4914 (0.1056)
IAC	0.2265 (0.0056)	0.1691 (0.0056)	0.1690 (0.0056)	0.2144 (0.0063)
IAC-Fit	0.0363 (0.0002)	0.0282 (0.0002)	0.0262 (0.0002)	0.0347 (0.0002)

Table 5: Mean Euclidean distance between the empirical data set and different statistical cultures (standard error in parentheses) for elections with 3 candidates.

	IC	DC	UC	UUP
Set 1	0.2394 (0.0046)	0.1967 (0.0031)	0.0991 (0.0020)	0.2533 (0.0120)
Set 2	0.2379 (0.0064)	0.1931 (0.0042)	0.0975 (0.0023)	0.2491 (0.0127)
Set 3	0.2633 (0.0079)	0.2129 (0.0051)	0.1153 (0.0032)	0.2902 (0.0159)
Set 4	0.2623 (0.0069)	0.2156 (0.0039)	0.1119 (0.0035)	0.2767 (0.0169)
Set 5	0.2458 (0.0044)	0.2040 (0.0028)	0.1059 (0.0027)	0.2633 (0.0138)
Set 6	0.3046 (0.0077)	0.2443 (0.0045)	0.1214 (0.0040)	0.3209 (0.0223)
Set 7	0.2583 (0.0088)	0.2094 (0.0053)	0.1060 (0.0038)	0.2710 (0.0161)
Set 8	0.2573 (0.0052)	0.2095 (0.0034)	0.1059 (0.0023)	0.2508 (0.0145)
Set 9	0.2981 (0.0090)	0.2414 (0.0049)	0.1202 (0.0045)	0.3258 (0.0241)
Set 10	0.2223 (0.0046)	0.1791 (0.0035)	0.1053 (0.0021)	0.2327 (0.0085)
Urn	0.6599 (0.0201)	0.4744 (0.0126)	0.4745 (0.0126)	0.6564 (0.1022)
IAC	0.1258 (0.0004)	0.0899 (0.0004)	0.0900 (0.0004)	0.1274 (0.0004)
IAC-Fit	0.0463 (0.0001)	0.0340 (0.0001)	0.0318 (0.0001)	0.0472 (0.0001)

Table 6: Mean Euclidean distance between the empirical data set and different statistical cultures (standard error in parentheses) for elections with 4 candidates.

Table 5 and Table 6 summarizes our results for the analysis of different statistical models used to generate elections. In general, none of the probability models captures our empirical data. Uniform Culture (UC) has the lowest error in predicting the distributions found in our empirical data. We conjecture that this is due to the process by which we select movies and the fact that these are ratings on movies. Since we require strict orders and, generally, most people rate good movies better than bad movies, we obtain elections that look like UC scenarios. By this we mean that *The Godfather* is an objectively good movie while *Mega Shark vs. Crocosaurus* is pretty bad. While there are some people who may reverse these movies, most users will rate *The Godfather* higher. This gives the population something close to a UC when investigated in the way that we do here.

The data generated by our IAC-Fit model fits very closely to the various statistical models. This is most likely due to the fact that the distributions generated by the IAC-Fit procedure closely resemble an Impartial Culture (since our sample size is so large). We, like Tideman and Plassmann, find little support for the static cultures' ability to model real data [24]

5 Manipulation of Borda Elections

In this section, we present empirical results for experiments involving algorithms given by Zuckerman et al. to manipulate elections under the Borda voting rule [27]. Much of the analysis of manipulation and algorithms for manipulation takes place in the theoretical domain, including looking at the frequency of manipulation relative to the total election size for scoring rules given by Xia and Conitzer [26]. Additionally, Pritchard et al. have looked at the asymptotic and average set sizes necessary to manipulate elections under a variety of rules [18, 19]. Unfortunately, Pritchard's analysis is under the Impartial Culture assumption, which is an election distribution that we have seen does not match our data.

Our experiment takes ballot data for an election under the Borda rule and a non-winning candidate, then adds manipulators one by one until the distinguished candidate wins. The question we ask is, how many manipulators are needed? The algorithm greedily calculates the ballot for each manipulator, given all of the unmanipulated ballots and the ballots of the previous manipulators. The next manipulator's ballot has the distinguished candidate first, and then lists the rest of the candidates in reverse order of their total points so far [27]. This algorithm by Zuckerman et. al has been proven to either find the optimal coalitional manipulation, or over-guess by one voter [27]. In a furthur empirical study Davies et al. compared two additional algorithms for finding Borda manipulations to Zuckerman et al.'s [8]. Davies et al. found that, while all three algorithms found the optimal manipulation over 75% of the time, Davies et al.'s AVERAGE FIT algorithm found the optimal manipulation over 99% of the time.

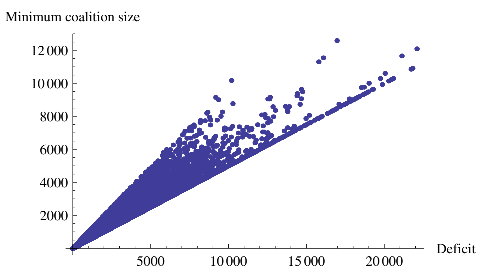


Figure 1: Deficit vs. minimum coalition size for Zuckerman's algorithm

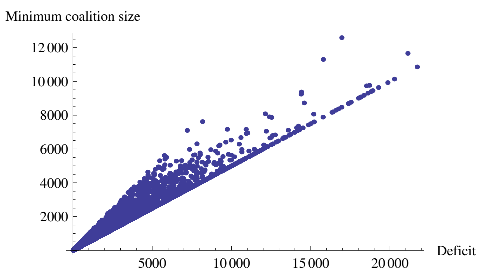


Figure 2: Deficit vs. minimum coalition for promoting third-place candidates

The size of the coalition is determined both by the distribution of votes and by the *deficit* of the distinguished candidate, namely, the difference between the number of points assigned to the current winner and the number of points assigned to the distinguished candidate. We ask a fundamentally different question than the earlier experiments on Borda manipulation algorithms. At minimum, a Borda manipulation requires a coalition size linear in the deficit size, d [8]. We want to know how often, and under what conditions, do we have a linear coalition requirement versus when we require a super-linear coalition.

Figure 1 shows the relationship of the initial deficit to the coalition size. For our experiment we used $296,553$ elections, ranging in size from 350 to 18,269 voters, from Set 1 (detailed in Section 3). The average number of voters per election in this size is 991.68, and the median is 621. Each point in the graph in Figure 1 represents the a coalition size for an election with that deficit, regardless of which candidate was promoted. For 99% of the elections we tested, it took $\lfloor \frac{d}{2} \rfloor + 1$ coalition members. Figure 2 shows coalition sizes as a function of deficit for promoting the third-place candidate to a winner.

For those elections where promoting the 3rd-place candidate took a coalition of more than $\lfloor \frac{d}{2} \rfloor + 1$, the average deficit for promoting the *second-place* candidate is 306, and the average corresponding coalition size is 154 ($= \lfloor \frac{d}{2} \rfloor + 1$). For those elections, the average deficit for promoting the *third-place* candidate is 873, and the average corresponding coalition size is 572.

6 Conclusion

We have identified and thoroughly evaluated a novel dataset as a source of sincere election data. We find overwhelming support for many of the existing conclusions in the empirical literature. Namely, we find a high consensus among a variety of voting methods; low occurrences of Condorcet's Paradox and other voting cycles; low occurrences of preference domain restrictions such as single-peakedness; a lack of support for existing statistical models which are used to generate election pseudo-data; and some interesting differences between the sizes of coalitions needed to promote a 2nd-place candidate and a 3rd-place candidate, using Zuckerman's algorithm for Borda. Our study is significant as it adds more results to the current discussion of what is an election and how often do voting irregularities occur? Voting is a common method by which agents make decisions both in computers and as a society. Understanding the unique statistical and mathematical properties of voting rules, as verified by empirical evidence across multiple domains, is an important step. We provide a new look at this question with a novel dataset that is several orders of magnitude larger than the sum of the data in previous studies. This empirical work is very much in the spirit of the overall ComSoc approach: we are using computational tools (data mining and access to extremely large sets of preference data) to address concerns in the social choice community. It is our hope that, with this dataset, we inspire others to look for novel datasets and empirically test some of their theoretical results.

The collection and public dissemination of the datasets is a central point our work. We plan to establish a repository of election data so that theoretical researchers can validate with empirical data. We plan to identify several other free, public datasets that can be viewed as "real world" voting data. The results reported in our study imply that our data is reusable as real world voting data. Therefore, it seems that the Netflix dataset, and its $> 10^{12}$ possible elections, can be used as a source of election data for future empirical validation of theoretical voting studies. We would like to, instead of comparing how voting rules correspond to one another, evaluate their power as maximum likelihood estimators [7]. Additionally, we would like to expand our evaluation of statistical models to include several new models proposed by Tideman and Plassmann, and others [24]. We will continue to analyze manipulation algorithms from the literature on elections from this data set.

Acknowledgements

Thanks to Dr. Florenz Plassmann for his helpful discussions on this paper and guidance on calibrating statistical models and to Elizabeth Mattei and Tom Dodson for their helpful discussion and comments on preliminary drafts of this paper. We thank the reviewers who have made this a better paper through their careful reviews. This work is supported by the National Science Foundation, under EAGER grant CCF-1049360. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] K. Arrow, A. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare*, volume 1. North-Holland, 2002.

[2] J. Bennett and S. Lanning. The Netflix Prize. In *Proceedings of KDD Cup and Workshop*, 2007. `www.netflixprize.com`.

[3] S. Berg. Paradox of voting under an urn model: The effect of homogeneity. *Public Choice*, 47(2):377–387, 1985.

[4] D. Black. On the rationale of group decision-making. *The Journal of Political Economy*, 56(1), 1948.

[5] F. Brandt, M. Brill, E. Hemaspaandra, and L. A. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI 2010)*, pages 715 – 722, 2010.

[6] J. R. Chamberlin, J. L. Cohen, and C. H. Coombs. Social choice observed: Five presidential elections of the American Psychological Association. *The Journal of Politics*, 46(2):479 – 502, 1984.

[7] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proc. of the 21st Annual Conf. on Uncertainty in AI (UAI)*, pages 145–152, 2005.

[8] J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. Complexity of and algorithms for borda manipulation. In *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI 2011)*, pages 657–662, 2011.

[9] D. S. Felsenthal, Z. Maoz, and Rapoport A. An empirical evaluation of six voting procedures: Do they really make any difference? *British Journal of Political Science*, 23:1 – 27, 1993.

[10] W. V. Gehrlein. Condorcet's paradox and the likelihood of its occurance: Different perspectives on balanced preferences. *Theory and Decisions*, 52(2):171 – 199, 2002.

[11] J. Han and M. Kamber, editors. *Data Mining*. Morgan Kaufmann, 2006.

[12] T. Lu and C. Boutilier. Robust approximation and incremental elicitation in voting protocols. In *Proc. of the 22nd Intl. Joing Conf. on Artificial Intelligence (IJCAI 2011)*, 2011.

[13] N. Mattei. Empirical evaluation of voting rules with strictly ordered preference data. In *Proceedings of the 2nd International Conference on Algorithmic Decision Theory (ADT 2011)*, 2011.

[14] N. Mattei. *Decision Making Under Uncertainty: Theoretical and Empirical Results on Social Choice, Manipulation, and Bribery*. PhD thesis, University of Kentucky, 2012.

[15] S. Merrill, III. A comparison of efficiency of multicandidate electoral systems. *American Journal of Politial Science*, 28(1):23 – 48, 1984.

[16] R. G. Niemi. The occurrence of the paradox of voting in university elections. *Public Choice*, 8(1):91–100, 1970.

[17] H. Nurmi. Voting procedures: A summary analysis. *British Journal of Political Science*, 13:181 – 208, 1983.

[18] G. Pritchard and A. Slinko. On the average minimum size of a manipulating coalition. *Social Choice and Welfare*, 27(2):263–277, 2006.

[19] G. Pritchard and M.C. Wilson. Asymptotics of the minimum manipulating coalition size for positional voting rules under impartial culture behaviour. *Mathematical Social Sciences*, 58(1):35–57, 2009.

[20] M. Regenwetter, B. Grogman, A. A. J. Marley, and I. M. Testlin. *Behavioral Social Choice: Probabilistic Models, Statistical Inference, and Applications*. Cambridge Univ. Press, 2006.

[21] M. Regenwetter, A. Kim, A. Kantor, and M. R. Ho. The unexpected empirical consensus among consensus methods. *Psychological Science*, 18(7):629 – 635, 2007.

[22] R. L. Rivest and E. Shen. An optimal single-winner preferential voting system based on game theory. In V. Conitzer and J. Rothe, editors, *Proc. of the 3rd Intl. Workshop on Computational Social Choice (COMSOC 2010)*, pages 399 – 410, 2010.

[23] A. K. Sen. A possibility theorem on majority decisions. *Econometrica*, 34(2):491–499, 1966.

[24] N. Tideman and F. Plassmann. Modeling the outcomes of vote-casting in actual elections. In D.S. Felsenthal and M. Machover, editors, *Electoral Systems: Paradoxes, Assumptions, and Procedures*. Springer, 2012.

[25] T. Walsh. An empirical study of the manipulability of single transferable voting. In *Proc. of the 19th European Conf. on AI (ECAI 2010)*, pages 257–262. IOS Press, 2010.

[26] L. Xia and V. Conitzer. Generalized scoring rules and the frequency of coalitional manipulability. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 109–118. ACM, 2008.

[27] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392 – 412, 2009.

Nicholas Mattei, James Forshee, and Judy Goldsmith
Department of Computer Science
University of Kentucky
Lexington, KY 40506, USA
Email: `nick.mattei@uky.edu,james.forshee@uky.edu,goldsmit@cs.uky.edu`

Bounding the Cost of Stability in Games with Restricted Interaction

Reshef Meir, Yair Zick, Edith Elkind, and Jeffrey S. Rosenschein

Abstract

We study stability of cooperative games with restricted interaction, in the model that was introduced by Myerson [20]. We show that the cost of stability of such games (i.e., the subsidy required to stabilize the game) can be bounded in terms of natural parameters of their interaction graphs. Specifically, we prove that if the treewidth of the interaction graph is k, then the relative cost of stability is at most $k + 1$, and this bound is tight for all $k \geq 2$. Also, we show that if the pathwidth of the interaction graph is k, then the relative cost of stability is at most k.

1 Introduction

Coalitional game theory models scenarios where groups of agents can work together profitably: the agents form teams, or *coalitions*, and each coalition generates a payoff, which then needs to be shared among the members of that coalition. The agents are assumed to be selfish, so the payoffs should be divided in such a way that each agent is satisfied with his share. In particular, it is desirable to allocate the payoffs so that no group of agents can do better by deviating from their current coalitions and embarking on a project of their own; the set of all payoff division schemes that have this property is known as the *core* of the game. However, this requirement turns out to be very strong: indeed, there are many games that have an empty core.

There are several ways to capture the intuition behind the notion of the core while relaxing the core constraints. For instance, one can assume that deviation comes at a cost, so players will not deviate unless the profit from doing so exceeds a certain threshold; formalizing this approach leads to the notions of *ε-core* and *least core*. Alternatively, one can assume that the deviators are non-myopic, and will not attempt a deviation if it may be followed by a counter-deviation that makes them worse off; this idea is captured by the notion of *bargaining set*. Yet another approach, which was pioneered by Myerson [20], is based on the idea that communication among agents may be limited, and agents cannot form a deviating coalition unless they can communicate with one another. In more detail, the communication network among the agents is described by an *interaction graph*, where agents are nodes, and an edge denotes the presence of a communication link; allowable coalitions correspond to connected subgraphs of the interaction graph. Myerson's model can be seen as a special case of a restriction scheme known as *partition systems* (see Chapter 5 in Bilbao [6] for an overview). Finally, coalitional stability may be achieved via *subsidies*: an external party may be willing to stabilize the game by offering a lump sum to the agents as long as they form some desired coalition structure. The minimal subsidy required in order to guarantee stability is known as the *cost of stability (CoS)* [4] (in what follows, it will be convenient to use a modified version of this notion known as *relative cost of stability ($RCoS$)* [19], which is defined as the ratio between the minimal total payoff needed to ensure stability and the total value of an optimal coalition structure).

In this paper, we study the interplay between the latter two concepts, namely, restricted interaction and the cost of stability. Our goal is to bound the (relative) cost of stability of a game in terms of natural parameters of its interaction graph. One such parameter is the *treewidth*: this is a combinatorial measure of graph structure that ranges from 1 (a tree or a forest) to $n - 1$ (a complete graph on n vertices), and, intuitively, says whether the graph is close to being a tree. A closely related notion is that of *pathwidth*, which measures how close the graph is to being a path. We are motivated by the classic result of Demange [10], who showed that if the interaction graph is a tree then the core of the game is not empty. Given this result, it is natural to ask if games whose interaction graphs have small treewidth are close to having a non-empty core.

Our Contribution Our main contribution is a complete characterization of the relationship between the treewidth of the interaction graph and the cost of stability. We show that if the treewidth of the interaction graph of a game G is k, then the relative cost of stability of G is at most $k+1$. Moreover, we demonstrate that this bound is tight whenever $k \geq 2$. We also show that the bound on the relative cost of stability can be improved to k if the *pathwidth* of the interaction graph is k, and this is also tight.

Related Work There is a significant body of work on subsidies in cooperative games. Many of the earlier papers focused on *cost-sharing games*, where agents share the *cost* of a project, rather than its profits (see, for example, [17, 12]). For profit-sharing games, Bachrach et al. [4] have recently introduced the notion of cost of stability (CoS), which is defined as the minimal subsidy needed to stabilize such games. Bachrach et al. gave bounds on the cost of stability for several classes of coalitional games, and analyzed the complexity of computing the cost of stability in weighted voting games. Several groups of researchers have extended this analysis to other classes of coalitional games [21, 18, 2, 19, 14, 15]. In particular, Meir et al. [19] and Greco et al. [15] studied questions related to the CoS in games with restricted cooperation, providing bounds on the CoS for some simple graphs.

It is well known that many graph-related problems that are computationally hard in the general case become tractable once the treewidth of the underlying graph is bounded by a constant (see, e.g., [9]). There are several graph-based representation languages for cooperative games, and for many of them the complexity of computational questions that arise in cooperative game theory (such as finding an outcome in the core or an optimal coalition structure) has been bounded in terms of the treewidth of the corresponding graph [16, 3, 5, 14]. However, in general bounding the treewidth of the Myerson graph (except for the special case of width 1) *does not* lead to a tractable solution for these computational questions, as shown by Greco et al. [15] and more recently by Chalkiadakis et al. [8]. Moreover, the notion of treewidth was mostly applied in the context of algorithmic analysis of cooperative games; to the best of our knowledge, our work is the first to employ treewidth to prove a game-theoretic result that is not computational in nature.

2 Preliminaries

We will now present the definitions that will be used in this paper. In what follows, we use boldface lowercase letters to denote vectors, and uppercase letters to denote sets of agents.

A *transferable utility (TU) game* is a tuple $G = \langle N, v\rangle$, where $N = \{1, \dots, n\}$ is a finite set of *agents* and $v : 2^N \to \mathbb{R}$ is the *characteristic function* of the game. We assume that $v(\emptyset) = 0$. Also, unless explicitly stated otherwise, we restrict our attention to games where the characteristic function takes non-negative values only, i.e., $v(S) \geq 0$ for all $S \subseteq N$.

A TU game $G = \langle N, v\rangle$ is *superadditive* if $v(S \cup T) \geq v(S) + v(T)$ for every $S, T \subseteq N$ such that $S \cap T = \emptyset$; it is *monotone* if $v(S) \leq v(T)$ for every $S, T \subseteq N$ such that $S \subseteq T$. Further, G is said to be *simple* if for all $S \subseteq N$ it holds that $v(S) \in \{0, 1\}$. Note that, unlike, e.g., [22], we *do not* require simple games to be monotone; this allows us to use the inductive argument in Section 3.2. A coalition S in a simple game $G = \langle N, v\rangle$ is said to be *winning* if $v(S) = 1$ and *losing* if $v(S) = 0$.

Following [1], we assume that agents may form coalition structures. A *coalition structure* over N is a partition of N into disjoint subsets. The *value* of a coalition structure CS over N, denoted by $v(CS)$, is given by $v(CS) = \sum_{S\in CS} v(S)$. We denote the set of all coalition structures over N by $\mathcal{CS}(N)$, and write $OPT(G) = \max\{v(CS) \mid CS \in \mathcal{CS}(N)\}$. CS is said to be *optimal* if $v(CS) = OPT(G)$. Note that in superadditive games $v(N) = OPT(G)$.

Payoffs and Stability Having split into coalitions and generated profits, agents need to divide the gains among themselves. A *payoff vector* is simply a vector $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n_+$, where the i-th coordinate is the payoff to agent $i \in N$. We denote the total payoff to a set $S \subseteq N$ by $x(S)$, i.e., we write $x(S) = \sum_{i \in S} x_i$. We say that a payoff vector $\mathbf{x}$ is a *pre-imputation* for a coalition structure CS if for all $S \in CS$ it holds that $x(S) = v(S)$. A pair of the form $(CS, \mathbf{x})$, where $CS \in \mathcal{CS}(N)$ and $\mathbf{x}$ is a pre-imputation for CS, is referred to as an *outcome* of the game $G = \langle N, v \rangle$; an outcome is *individually rational* if $x_i \geq v(\{i\})$ for every $i \in N$. If $\mathbf{x}$ is a pre-imputation for CS that is individually rational, it is called an *imputation* for CS. We say that an outcome $(CS, \mathbf{x})$ of a game $G = \langle N, v \rangle$ is *stable* if $x(S) \geq v(S)$ for all $S \subseteq N$. The set of all stable outcomes of G is called the *core* of G, and is denoted $Core(G)$. We let $\mathcal{S}(G)$ denote the set of all payoff vectors (not necessarily pre-imputations) that satisfy the stability constraints, i.e., we set

$$\mathcal{S}(G) = \{\mathbf{x} \in \mathbb{R}^n \mid x(S) \geq v(S) \text{ for all } S \subseteq N\}.$$

We refer to payoff vectors such that $x(N) \geq OPT(G)$ as *super-imputations*; note that $\mathcal{S}(G)$ consists of super-imputations only.

The *Relative Cost of Stability* of a game G is the minimal total payoff that stabilizes the game. Formally, we set

$$RCoS(G) = \inf \left\{ \frac{x(N)}{OPT(G)} \mid \mathbf{x} \in \mathcal{S}(G) \right\}.$$

Note that $RCoS(G) \geq 1$ for every TU game G, and $RCoS(G) = 1$ implies $Core(G) \neq \emptyset$.

Interaction Graphs and Treewidth An *interaction network* over N is a graph $H = \langle N, E \rangle$. Given a game $G = \langle N, v \rangle$ and an interaction network over N, we define a game $G|_H = \langle N, v|_H \rangle$ by setting $v|_H(S) = v(S)$ if S forms a connected subgraph of H, and $v|_H(S) = 0$ otherwise; that is, in $G|_H$ a coalition $S \subseteq N$ may form if and only if S forms a connected subgraph of H.

A *tree decomposition* of H is a tree $\mathcal{T}$ over the nodes $V(\mathcal{T})$ with the following properties:

1. Each node of $\mathcal{T}$ is a subset of N.
2. For every pair of nodes $X, Y \in V(\mathcal{T})$ and every $i \in N$, if $i \in X$ and $i \in Y$ then for any node Z on the (unique) path between X and Y in $\mathcal{T}$ we have $i \in Z$.
3. For every edge $e = \{i, j\}$ of E there exists a node $X \in V(\mathcal{T})$ such that $e \subseteq X$.[1]

The *width* of a tree decomposition $\mathcal{T}$ is $tw(\mathcal{T}) = \max_{X \in V(\mathcal{T})} |X| - 1$; the *treewidth* of H is defined as $tw(H) = \min\{tw(\mathcal{T}) \mid \mathcal{T} \text{ is a tree decomposition of } H\}$. Examples of graphs with low treewidth include trees (whose treewidth is 1) and series-parallel graphs (whose treewidth is at most 2); see, e.g., [7].

Given a subtree $\mathcal{T}'$ of a tree decomposition $\mathcal{T}$ (we use the term "subtree" to refer to any connected subgraph of $\mathcal{T}$), we denote the agents that appear in the nodes of $\mathcal{T}'$ by $N(\mathcal{T}')$. Conversely, given a set of agents $S \subseteq N$, we let $\mathcal{T}(S)$ denote the subgraph of $\mathcal{T}$ induced by the node set $\{X \in V(\mathcal{T}) \mid X \cap S \neq \emptyset\}$; it is not hard to check that $\mathcal{T}(S)$ is a subtree of $\mathcal{T}$ for every $S \subseteq N$. Given a tree decomposition $\mathcal{T}$ of H and a node $R \in V(\mathcal{T})$, we can set R to be the root of $\mathcal{T}$. In this case, we denote the subtree rooted in a node $S \in V(\mathcal{T})$ by $\mathcal{T}_S$.

A tree decomposition of a graph H such that $\mathcal{T}$ is a path is called a *path decomposition* of H. The *pathwidth* of H is defined as $pw(H) = \min\{tw(\mathcal{T}) \mid \mathcal{T} \text{ is a path decomposition of } H\}$. It is known that for any graph H, $tw(H) \leq pw(H)$ and $pw(H) = tw(H) \cdot O(\log(n))$.

3 Treewidth and the Cost of Stability

Our goal in this section is to provide a general upper bound on the cost of stability for TU games whose interaction networks have bounded treewidth. We start by proving a bound for simple games; we then show how to extend it to the general case.

[1] We note that a tree decomposition of *hypergraphs* is defined in the same way, except that every *hyperedge* must be contained in some node.

3.1 Simple Games

Algorithm 1: STABLE-PAYOFF-TW($G = \langle N, v\rangle, H, k, \mathcal{T}$)

Fix an arbitrary $R \in V(\mathcal{T})$ to be the root;
$t \leftarrow 0$, $N_1 \leftarrow N$, $\mathbf{x} \leftarrow 0^n$;
for $A \in V(\mathcal{T})$*, traversed from the leaves upwards* **do**
  $t \leftarrow t + 1$;
  if *there is some* $S \subseteq N(\mathcal{T}_A) \cap N_t$ *such that* $v|_H(S) = 1$ **then**
    for $i \in A \cap N_t$ **do**
      $x_i \leftarrow 1$
    $N_{t+1} \leftarrow N_t \setminus N(\mathcal{T}_A)$;
    `// remove all agents in` $N(\mathcal{T}_A)$ `from the entire tree`
  else
    $N_{t+1} \leftarrow N_t$;
return $\mathbf{x} = (x_1, \ldots, x_n)$;

We will now show that if G is a game with a set of agents N and H is an interaction network over N then $RCoS(G|_H) \leq tw(H)+1$. Our proof is constructive: we design an algorithm (Algorithm 1) that receives as its input a simple game $G = \langle N, v\rangle$, a network H, a parameter k, and a tree decomposition $\mathcal{T}$ of H of width of at most k, and outputs a stable super-imputation for $G|_H$. Briefly, Algorithm 1 picks an arbitrary node $R \in V(\mathcal{T})$ to be the root of $\mathcal{T}$ and traverses the nodes of $\mathcal{T}$ from the leaves towards the root. Upon arriving at a node A, it checks whether the subtree $\mathcal{T}_A$ rooted in A contains a coalition that is winning in $G|_H$ (note that we have to check every subset of $N(\mathcal{T}_A) \cap N_t$, since $G|_H$ is not necessarily monotone). If this is the case, it pays 1 to all agents in A and removes all agents in $\mathcal{T}_A$ from every node of $\mathcal{T}$. Note that every winning coalition in $\mathcal{T}_A$ has to be connected, so either it is fully contained in a proper subtree of $\mathcal{T}_A$ or it contains agents in A. The reason for deleting the agents in $\mathcal{T}_A$ is simple: every winning coalition that contains members of $\mathcal{T}_A$ is already stable (one of its members is getting a payoff of 1). The algorithm then continues up the tree in the same manner until it reaches the root. Note that Algorithm 1 is very similar to the one proposed by Demange [10]; however, Algorithm 1 may pay $2 \cdot OPT(G|_H)$ if H is a tree.[2] Moreover, while Demange's algorithm runs in polynomial time, Algorithm 1 may require exponential time, since it is designed to work for non-monotone simple games. However, if the simple game given as input is monotone, a straightforward modification (check whether $v|_H(S) = 1$ only for $S = N(\mathcal{T}_A)$ rather than for every $S \subseteq N(\mathcal{T}_A)$) will make it run in polynomial time.

Theorem 3.1. *For every simple game $G = \langle N, v\rangle$ and every interaction network H over N it holds that $RCoS(G|_H) \leq tw(H) + 1$.*

Proof. Let $\mathcal{T}$ be a tree decomposition of H such that $tw(\mathcal{T}) = k$. Suppose first that $G|_H$ is superadditive. This means that any two winning coalitions in $G|_H$ intersect. Hence, for every pair of winning coalitions $S_1, S_2 \subseteq N$ the subtrees $\mathcal{T}(S_1)$ and $\mathcal{T}(S_2)$ intersect. This implies that there exists a node $A \in V(\mathcal{T})$ that belongs to the intersection of all subtrees that correspond to winning coalitions in $\mathcal{T}$ (this fact is known as Helly's Theorem for Trees), and hence intersects every winning coalition. Therefore we can stabilize the game by paying 1 to every agent in A. Thus, our total payment is $|A| \leq tw(\mathcal{T}) + 1 \leq k + 1$.

We now turn to the more general case of arbitrary simple games. Let $\mathbf{x}$ be the output of Algorithm 1. We claim that $\mathbf{x}$ is stable (i.e., $\mathbf{x} \in \mathcal{S}(G|_H)$) and $x(N) \leq k + 1$.

[2]This is because Algorithm 1 operates on the tree decomposition $\mathcal{T}$ of H, which has nodes of size 2. In this special case we can modify our algorithm by only paying one of the agents in A—the one that does not appear above A in the tree. The resulting payoff vector would then coincide with the one constructed by Demange's algorithm.

To prove stability, consider a coalition S with $v|_H(S) = 1$; we need to show that $x(S) > 0$. Suppose for the sake of contradiction that $x(S) = 0$; this means that each agent in S is deleted before he is allocated any payoff. Consider the first time step when an agent in S is deleted; suppose that this happens at step t when a node $A \in V(\mathcal{T})$ is processed. Clearly for an agent in S to be deleted at this step it has to be the case that $\mathcal{T}(S) \cap \mathcal{T}_A \neq \emptyset$. Further, it cannot be the case that $S \cap (A \cap N_t) \neq \emptyset$, since each agent in $A \cap N_t$ is assigned a payoff of 1 at step t, and we have assumed that $x(S) = 0$. Therefore, $\mathcal{T}(S)$ must be a proper subtree of $\mathcal{T}_A$. Let B be the root of $\mathcal{T}(S)$, and consider the time step $t' < t$ when B is processed. At time t', all agents in S are still present in $\mathcal{T}$, so the node B meets the **if** condition in Algorithm 1, and therefore each agent in B gets assigned a payoff of 1. This is a contradiction, since B is the root of $\mathcal{T}(S)$, and therefore $B \cap S \neq \emptyset$, which implies $x(S) > 0$.

It remains to show that $x(N) \leq (k+1)OPT(G)$. To this end, we will construct a specific coalition structure CS^* and argue that $x(N) \leq (k+1)v(CS^*)$.

The coalition structure CS^* is constructed as follows. Let A_t be the node of the tree considered by Algorithm 1 at time t, and let $S_t = N(\mathcal{T}_{A_t}) \cap N_t$, i.e., S_t is the set of all agents that appear in $\mathcal{T}_{A_t}$ at time t. Let T^* be the set of all values of t such that A_t meets the **if** condition in Algorithm 1. For each $t \in T^*$ the set S_t contains a winning coalition; let W_t be an arbitrary winning coalition contained in S_t. Finally, let $L = N \setminus (\cup_{t \in T^*} W_t)$, and set

$$CS^* = \{L\} \cup \{W_t \mid t \in T^*\}.$$

Observe that CS^* is a coalition structure, i.e., a partition of N. Indeed, $L \cap W_t = \emptyset$ for all $t \in T^*$, and, moreover, if $i \in W_t$ for some $t > 0$, then i was removed from $\mathcal{T}$ at time t, and cannot be a member of coalition $W_{t'}$ for $t' > t$. Further, we have $v(CS^*) = |T^*|$.

To bound the total payment, we observe that no agent is assigned any payoff at time $t \notin T^*$, and each agent that is assigned a payoff of 1 at time $t \in T^*$ is a member of A_t. Hence we have

$$\begin{aligned} x(N) &= \sum_{t \in T^*} x(A_t) \leq \sum_{t \in T^*} |A_t| \leq \sum_{t \in T^*} (k+1) \\ &= (k+1)|T^*| = (k+1)v(CS^*) \leq (k+1)OPT(G), \end{aligned}$$

which proves that $RCoS(G) \leq k+1$. □

We note that under the payment scheme constructed by Algorithm 1 the payoff of every agent is either 1 or 0. Note also that the proof of Theorem 3.1 goes through as long as $G|_H$ is simple, even if G itself is not simple.

3.2 The General Case

Using Theorem 3.1, we are now ready to prove our main result.

Theorem 3.2. *For every game $G = \langle N, v \rangle$ and every interaction network H over N it holds that $RCoS(G|_H) \leq tw(H) + 1$.*

Proof. We first prove the claim for all integer-valued games. We use an inductive argument on $OPT(G|_H) = m$. If $OPT(G|_H) = 1$ then in particular $G|_H$ is simple, so we are done by Theorem 3.1. Now suppose that our claim is true for all $m' < m$; we will show that it holds for m. To simplify notation, we identify v with $v|_H$, i.e., we write v in place of $v|_H$ throughout the proof. We define the following simple game $G' = \langle N, v' \rangle$:

$$v'(S) = \begin{cases} 1 & \text{if } v(S) > 0 \\ 0 & \text{otherwise} \end{cases}$$

By Theorem 3.1, there exists a super-imputation $\mathbf{x}'$ such that $x'(S) \geq v'(S)$ for all $S \subseteq N$ and $x'(N) \leq (tw(H)+1)v(CS')$, where CS' is an optimal coalition structure over G'. Moreover, we can assume that $\mathbf{x}' \in \{0,1\}^n$, as Algorithm 1 outputs such a super-imputation. We define a game $G'' = \langle N, v''\rangle$ by setting

$$v''(S) = \max\{0, v(S) - x'(S)\}.$$

Note that $v''(S) \in \mathbb{Z}^+$ for all $S \subseteq N$, since $\mathbf{x}' \in \{0,1\}^n$ and G is integer-valued. Moreover, let CS'' be an optimal coalition structure for G'', and let $CS''_+ = \{S \in CS'' \mid v''(S) > 0\}$. We have

$$\sum_{S \in CS''} v''(S) = \sum_{S \in CS''_+} v''(S) = \sum_{S \in CS''_+} v(S) - x'(S).$$

Moreover, for every $S \in CS''_+$ we have $v(S) - x'(S) > 0$; in particular this means that $v(S) > 0$, which implies that $v'(S) = 1 \leq x'(S)$. Therefore for any $S \in CS''_+$ we have

$$v''(S) = v(S) - x'(S) \leq v(S) - 1 < v(S).$$

We conclude that

$$\sum_{S \in CS''} v''(S) = \sum_{S \in CS''_+} v''(S) < \sum_{S \in CS''_+} v(S) \leq m.$$

Thus, the value of an optimal coalition structure over G'' is strictly less than m, i.e., we can apply the induction hypothesis to G''. This means that there is a super-imputation $\mathbf{x}''$ such that $x''(N) \leq (tw(H)+1)v''(CS'')$ and $x''(S) \geq v''(S)$ for all $S \subseteq N$. We set $\mathbf{x} = \mathbf{x}' + \mathbf{x}''$. We will now show that $x(N) \leq (tw(H)+1)\,OPT(G)$ and $x(S) \geq v(S)$ for all $S \subseteq N$.

First, observe that for all $S \subseteq N$ we have $x(S) = x'(S) + x''(S) \geq x'(S) + v''(S) \geq x'(S) + v(S) - x'(S) = v(S)$, so $\mathbf{x}$ is a stable super-imputation for G. Now, let CS'' be an optimal coalition structure over G'', and consider $CS'' \setminus CS''_+$, i.e., the set of all coalitions of value 0 in CS''. We can assume without loss of generality that $CS'' \setminus CS''_+$ is a singleton, i.e., there is only one coalition of value 0 in CS''; we denote this coalition by S_0. Let CS' be an optimal coalition structure over G', and let $CS'_+ = \{S \in CS' \mid v'(S) = 1\}$. Set $N^* = N \setminus S_0$; then we have

$$x'(N^*) \geq \sum_{S \in CS'_+} x'(S \cap N^*) \geq \sum_{S \in CS'_+} v'(S \cap N^*) = |\{S \in CS'_+ \mid S \cap N^* \neq \emptyset\}|.$$

Let $t^* = |\{S \in CS'_+ \mid S \cap N^* \neq \emptyset\}|$ and let $t_0 = |\{S \in CS'_+ \mid S \subseteq S_0\}|$. t^* is number of coalitions in CS'_+ that intersect N^*, and t^* is the number of those that are contained in S_0. The total value of CS' is thus $|CS'_+| = t^* + t_0$.

We are now ready to bound $x(N)$. We obtain

$$\begin{aligned}
x(N) &= x'(N) + x''(N) \leq (tw(H)+1)v''(CS'') + (tw(H)+1)v'(CS') \\
&= (tw(H)+1)\left(\sum_{S \in CS''_+} (v(S) - x'(S)) + |CS'_+|\right) \\
&= (tw(H)+1)\left(\sum_{S \in CS''_+} v(S) - x'(N^*) + |CS'_+|\right) \\
&\leq (tw(H)+1)\left(v(CS''_+) - t^* + |CS'_+|\right) = (tw(H)+1)\left(v(CS''_+) + t_0\right). \qquad (1)
\end{aligned}$$

Further, we have $t_0 = \sum_{S \in CS'_+ : S \subseteq S_0} v'(S) \leq \sum_{S \in CS'_+ | S \subseteq S_0} v(S)$, so the final term in (1) is at most $(tw(H)+1)\left(v(CS''_+) + \sum_{S \in CS'_+ : S \subseteq S_0} v(S)\right)$. This is a sum over a partition of (a subset of) N, so its total value is at most that of $OPT(G|_H)$, which concludes the proof for the integer case.

To extend this result to non-integer-valued games, we make the following observation. Given a game $G = \langle N, v\rangle$, we can consider the game $\varepsilon G = \langle N, v_\varepsilon\rangle$ given by $v_\varepsilon(S) = \varepsilon v(S)$ for every $S \subseteq N$; we note that if G is simple, then for any $\varepsilon > 0$ Algorithm 1 can be applied to the game εG and hence Theorem 3.1 remains true for εG. Moreover, in εG every agent receives a payoff of either ε or 0. Further, when defining the modified characteristic function v', we can set $\varepsilon = \min_{S\subseteq N}\{v(S) \mid v(S) > 0\}$ and let $v'(S) = \varepsilon$ whenever $v(S) > 0$ (instead of setting $v'(S) = 1$). The rest of the proof can be modified appropriately (with a different ε chosen at each iteration); in particular, instead of using induction on $OPT(G|_H)$, we use induction on the number of coalitions with non-zero value. □

The $RCoS$ of any cooperative game, even with unrestricted cooperation, is at most $\sqrt{n}$ (see [4, 18]). Thus, we obtain $RCoS(G|_H) \leq \min\{tw(H) + 1, \sqrt{n}\}$, assuming that coalition structures are allowed. Moreover, when applied to superadditive games, Theorem 3.2 implies that there is some stable super-imputation $\mathbf{x}$ such that $x(N) \leq (tw(H) + 1)v(N)$.[3]

Finally, since a simple superadditive game can be viewed as a collection of intersecting sets, we obtain the following corollary, which may be of independent interest.

Corollary 3.3. *Let $H = \langle N, E\rangle$ be a graph, and let $R_k = \langle N, \mathcal{F}, k\rangle$ be an instance of* HITTING SET *[13], where $\mathcal{F} = \{S_j\}_{j=1}^m$ is a collection of pairwise intersecting subsets of N, and every S_j is connected in H (i.e., $\langle S_j, E|_{S_j}\rangle$ is connected). Then for all $k \leq tw(H) - 1$ it holds that R_k is a "yes"-instance of* HITTING SET *and a hitting set of size (at most) k can be found efficiently.*

3.3 Tightness

Demange [10] showed that if $tw(H) = 1$, then the game $G|_H$ admits a stable outcome, i.e., $RCoS(G|_H) = 1$. This result is limited to games whose interaction networks are trees. However, we will now show that if the treewidth of the interaction network is at least 2, then the upper bound of $tw(H) + 1$ proved in Theorem 3.2 is tight.

Theorem 3.4. *For every $k \geq 2$ there is a simple superadditive game $G = \langle N, v\rangle$ and an interaction network H over N such that $tw(H) = k$ and $RCoS(G|_H) = k + 1$.*

Proof. Instead of defining H directly, we will describe its tree decomposition $\mathcal{T}$. There is one central node $A = \{z_1, \dots, z_{k+1}\}$. Further, for every unordered pair $I = \{i, j\}$, where $i, j \in \{1, \dots, k+1\}$ and $i \neq j$, we define a set D_I that consists of 7 agents and set $N = A \cup \bigcup_{i\neq j\in\{1,\dots,k+1\}} D_{\{i,j\}}$.

The tree $\mathcal{T}$ is a star, where leaves are all sets of the form $\{z_i, z_j, d\}$, where $d \in D_{\{i,j\}}$. That is, there are $7\cdot\binom{k+1}{2}$ leaves, each of size 3. Since the maximal node of $\mathcal{T}$ is of size $k+1$, it corresponds to some network whose treewidth is at most k. We set $\mathcal{D}_i = \bigcup_{j\neq i} D_{\{i,j\}}$; observe that for any two agents $z_i, z_j \in A$ we have $\mathcal{D}_i \cap \mathcal{D}_j = D_{\{i,j\}}$. Given $\mathcal{T}$, it is now easy to construct the underlying interaction network H: there is an edge between z_i and every $d \in D_{\{i,j\}}$ for every $j \neq i$; see Figure 1 for more details.

For every unordered pair $I = \{i, j\} \subseteq \{1, \dots, k + 1\}$, let $\mathcal{Q}_I$ denote the projective plane of dimension 3 (a.k.a. the Fano plane) over D_I. That is, $\mathcal{Q}_I$ contains seven triplets of elements from D_I, so that every two triplets intersect, and every element $d \in D_I$ is contained in exactly 3 triplets in $\mathcal{Q}_I$. Winning sets are defined as follows. For every $i = 1, \dots, k + 1$ and every selection $\{Q_{\{i,j\}} \in \mathcal{Q}_{\{i,j\}}\}_{j\neq i}$ the set $\{z_i\} \cup \bigcup_{j\neq i} Q_{\{i,j\}}$ is winning. Thus for every z_i there are 7^k winning coalitions containing z_i, each of size $1 + 3k$. Let us denote by $\mathcal{W}_i$ the set of winning coalitions that contain z_i; observe that for every $d \notin A$, d appears in exactly $3 \cdot 7^{k-1}$ winning coalitions in $\mathcal{W}_i$: d belongs to some $D_{\{i,j\}}$, and is selected to be in a winning coalition with z_i if a triplet $Q_{\{i,j\}}$

[3]Note that, while the proof for *simple* superadditive games is straightforward, we cannot use the inductive argument made in Theorem 3.2 directly, as superadditivity may not be preserved; therefore, we must go through all steps of the proof.

containing d is joined to z_i. There are 3 triplets in $\mathcal{Q}_{\{i,j\}}$ that contain d, and there are 7^{k-1} ways to choose the other triplets (seven choices from every one of the other $k-1$ sets).

We first argue that all winning coalitions intersect. Indeed, let C_i, C_j be winning coalitions such that $z_i \in C_i, z_j \in C_j$. Then both C_i and C_j contain some triplet from $\mathcal{Q}_{\{i,j\}}$. Suppose $Q_{\{i,j\}} \subseteq C_i, Q'_{\{i,j\}} \subseteq C_j$. Since $Q_{\{i,j\}}, Q'_{\{i,j\}} \in \mathcal{Q}_{\{i,j\}}$, they must intersect, and thus C_i and C_j must also intersect. This implies that the simple game induced by these winning coalitions is indeed superadditive and has an optimal value of 1. Note that if we pay 1 to each $z_i \in A$, then the resulting super-imputation is stable, since every winning coalition intersects A. To conclude the proof, we must show that any stable super-imputation must pay at least $k+1$ to the agents.

Given a stable super-imputation $\mathbf{x}$, we know that $x(C_i) \geq 1$ for every $C_i \in \mathcal{W}_i$. Thus, $\sum_{C_i \in \mathcal{W}_i} x(C_i) \geq 7^k$. We can write $\sum_{C_i \in \mathcal{W}_i} x(C_i)$ as

$$\begin{aligned}
\sum_{C_i \in \mathcal{W}_i} x(C_i) &= \sum_{C_i \in \mathcal{W}_i} \left(x_{z_i} + \sum_{d \neq z_i | d \in C_i} x_d \right) = 7^k x_{z_i} + \sum_{C_i \in \mathcal{W}_i} \sum_{d \neq z_i | d \in C_i} x_d \\
&= 7^k x_{z_i} + \sum_{d \in \mathcal{D}_i} 1 \sum_{C_i \in \mathcal{W}_i | d \in C_i} x_d = 7^k x_{z_i} + \sum_{d \in \mathcal{D}_i} 3 \cdot 7^{k-1} x_d \\
&= 7^k x_{z_i} + 3 \cdot 7^{k-1} x(\mathcal{D}_i).
\end{aligned}$$

This immediately implies that $x_{z_i} \geq 1 - \frac{3}{7} x(\mathcal{D}_i)$. Observe that $\sum_{z_i \in A} x(\mathcal{D}_i) = 2 \sum_{i<j} x(D_{\{i,j\}})$, as each $D_{\{i,j\}}$ appears exactly twice in the summation: once in $\mathcal{D}_i$ and once in $\mathcal{D}_j$. Also, observe that $\sum_{i<j} x(D_{\{i,j\}}) = x(N \setminus A)$, so $\sum_{i=1}^{k+1} x(\mathcal{D}_i) = 2x(N \setminus A)$. Finally,

$$\begin{aligned}
x(N) &= x(A) + x(N \setminus A) = \sum_{i=1}^{k+1} x_{z_i} + x(N \setminus A) \\
&\geq \sum_{i=1}^{k+1} \left(1 - \frac{3}{7} x(\mathcal{D}_i) \right) + x(N \setminus A) = \sum_{i=1}^{k+1} 1 - \frac{3}{7} 2x(N \setminus A) + x(N \setminus A) \\
&= k + 1 + (1 - \frac{6}{7}) x(N \setminus A) \geq k + 1
\end{aligned}$$

Thus, the relative cost of stability in our game is at least $k+1$. □

We observe that Theorem 3.4 does not hold when $k = 1$ since the width of our construction is at least 2 (each leaf is of size 3). Indeed, if it were to hold for $k = 1$, we would obtain a contradiction with Demange's theorem.

4 Pathwidth and the Cost of Stability

For some graphs we can bound not just their treewidth, but also their pathwidth. For example, for a simple cycle graph both the treewidth and the pathwidth are equal to 2. For games over interaction networks with bounded pathwidth, the bound of $tw(H) + 1$ shown in Section 3 can be tightened.

Theorem 4.1. *For every TU game $G = \langle v, N \rangle$ and every interaction network H over N it holds that $RCoS(G|_H) \leq pw(H)$, and this bound is tight.*

Proof. Note first that it suffices to show that our bound holds for simple games; we can then use the reduction described in the proof of Theorem 3.2. For simple games, our proof is very similar to the proof of Theorem 3.1; however, here we will show that in every node A_j that satisfies the **if** condition of Algorithm 2 we can identify an agent that we do not need to pay.

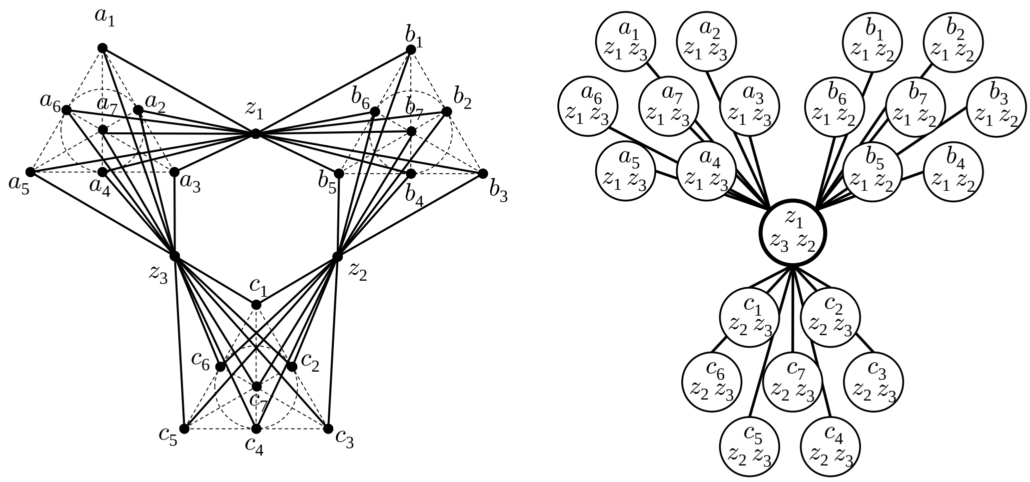


Figure 1: The interaction network H when $k = 2$ in Theorem 3.4. On the right there is the tree decomposition $\mathcal{T}$. There are three sets: $A = D_{1,3} = \{a_1, \ldots, a_7\}$, $B = D_{1,2} = \{b_1, \ldots, b_7\}$ and $C = D_{2,3} = \{c_1, \ldots, c_7\}$. An edge connects z_1 to all agents in A and B, z_2 to B and C, and z_3 to C and A. Agent z_1 forms winning coalitions with triplets of agents from A and B that are on a dotted line, Similarly, z_2 and z_3 form winning coalitions with their respective sets.

Our algorithm first deals with winning coalitions of size 1. This step can be justified as follows. Suppose we remove all agents in $I = \{i \in N \mid v(\{i\}) = 1\}$ and construct a stable super-imputation $\mathbf{x}'$ for the game $G'|_H$, where $G' = \langle N', v' \rangle$, $N' = N \setminus I$, and $v'(S) = v(S)$ for each $S \subseteq N \setminus I$, so that $x'(N') \leq pw(H)$. Now, consider a super-imputation $\mathbf{x}$ for G given by $x_i = 1$ for $i \in I$, $x_i = x'_i$ for $i \in N'$. We have $x(N) = x'(N') + |I|$, and, furthermore, $x(S) \geq v|_H(S)$ for every $S \subseteq N$, i.e., $\mathbf{x}$ is a stable super-imputation for $G|_H$. On the other hand, it is not hard to check that $OPT(G|_H) = OPT(G'|_H) + |I|$. Hence, we obtain

$$\frac{x(N)}{OPT(G|_H)} = \frac{x'(N') + |I|}{OPT(G'|_H) + |I|} < \frac{x'(N')}{OPT(G'|_H)} \leq pw(H),$$

i.e., $\mathbf{x}$ witnesses that $RCoS(G|_H) \leq pw(H)$. Thus, we begin Algorithm 2 by paying all winning singletons 1 and ignoring them (and any winning coalitions that contain them) for the rest of the execution; note, however, that we *do not* remove the winning singletons from H, i.e., we do not modify our path decomposition or its width.

Next we show stability. Given a node A_j, we must make sure that each winning coalition in $N(\mathcal{T}_{A_j})$ is paid at least 1. By the proof of Theorem 3.1, paying all agents in A_j is sufficient. Note, however, that there is no need to pay an agent i that is not in $N(\mathcal{T}_{A_j}) \setminus A_j$: since we removed all winning singletons, every winning coalition in $N(\mathcal{T}_{A_j})$ that contains i (and that is not yet stabilized) must also contain another agent from A_j.

Finally, we must show that in every paid node A_j, $j \geq 2$, there is at least one agent that is not paid. Note that A_j has a unique child A_{j-1}. If $A_j \subseteq A_{j-1}$, then no agent in A_j is being paid (as they had already been paid when processing A_{j-1}). Otherwise, there is some agent $i \in A_j \setminus A_{j-1}$. Since $\mathcal{T}$ is a path and all nodes containing i must be connected, we have $i \notin N(A_j) \setminus A_j$. Thus i is not paid. Note that in Algorithm 2 the agents in A_1 are not paid in the first iteration of the algorithm.

To show tightness, we use a slight modification of the construction from Section 3.3; we omit the details due to space constraints. □

Algorithm 2: STABLE-PAYOFF-PW($G = \langle N, v \rangle, H, k, \mathcal{T}$)

```
Set T = (A_1, ..., A_m);
x ← 0^n;
I ← {i ∈ N | v({i}) = 1};
for i ∈ I do
    x_i ← 1;
N_1 ← N \ I;
// Remove all singletons
t ← 1;
for j = 1 to m do
    if there is some S ⊆ N(T_{A_j}) ∩ N_j such that v(S) = 1 then
        for i ∈ A_j ∩ N_j do
            if i ∈ N(T_{A_j}) \ A_j then
                // Pay agents unless it is the first node they
                   appear in
                x_i ← 1
        N_{j+1} ← N_j \ N(T_{A_j});
        // Remove all agents in N(T_{A_j}) from the entire path
    else
        N_{j+1} ← N_j;
return x = (x_1, ..., x_n);
```

5 Conclusions, Discussion, and Future Work

Our main result shows a tight connection between the treewidth of an interaction network and the minimal subsidy required to stabilize a game played by the interacting agents: Simply put, as the interaction becomes "simpler", the game becomes easier to stabilize. To the best of our knowledge, this is the first time that the notion of treewidth is used to obtain results that are purely game-theoretic rather than algorithmic in nature.

While we provide bounds on $RCoS$ both in terms of the treewidth of the interaction network and in terms of its pathwidth, we view the former result as more significant than the latter: indeed, the result for the pathwidth only provides an improved bound when the pathwidth is exactly equal to the treewidth, which is quite uncommon.

Our results imply a separation between games whose interaction networks are acyclic, which have been shown to be stable [10], and other games. That is, treewidth of 1 implies $RCoS$ of 1, but for any higher value of treewidth, the $RCoS$ is somewhat higher. In particular, the result of Demange *is not* a special case of our theorem, although it can be proved using a very similar technique (i.e., by breaking the game into multiple simple games).

Games with implicit Myerson graphs While interaction networks have been introduced by Myerson as an external restriction independent of the value function, for some families of cooperative games this restriction is implicit in the game description. A prominent example is the class of *induced subgraph games* (ISG) proposed by Deng and Papadimitriou [11], where agents correspond to vertices of a graph, and the value of a coalition is the sum of weights of the edges between coalition members. Imposing the very same graph as an interaction network will preserve the value of any coalition in the game. Therefore, we can deduce a bound on the $RCoS$ of a given ISG directly from its description, by measuring the treewidth of its underlying graph. Other families that implicitly induce a Myerson graph are matching games and some variations of network flow games.

Hypergraphs Myerson's model can be generalized to *hypergraphs* rather than graphs [23]. Since our methods work with tree decompositions rather than the interaction networks themselves, they apply equally well to this case. Interestingly, the underlying hypergraph of a game defined via a *marginal contribution net* [16] also induces a Myerson (hyper)graph, which can in turn be used to bound the required subsidy.

The least core We remark that the cost of stability is closely related to another important notion of stability in cooperative games, namely, the least core; specifically, Meir et al. [19] show that the value of both the strong least core and the weak least core of a cooperative game can be bounded in terms of its additive cost of stability. Briefly, the value of the least core measures the dissatisfaction of coalitions in the "most stable" outcome, and is perhaps the most standard measure of stability in cooperative games. Our results, combined with those of [19], imply that any bound on the treewidth or pathwidth of the interaction graph translates into a bound on this important quantity. This provides further evidence that simple social interactions increase stability.

5.1 Future Work

While our bound on the cost of stability is tight in the worst case, it may be further improved by considering finer restrictions on the structure of the interaction network and/or the value function itself. More generally, we believe that this new connection between a well-studied graph parameter such as the treewidth and the stability properties of a related game is fascinating. We look forward to studying how such parameters can be used to reveal other hidden connections in both cooperative and non-cooperative game theory.

Acknowledgments

Meir and Rosenschein are partially supported by Israel Ministry of Science and Technology grant #3-6797, and the Google Inter-University Center for Electronic Markets and Auctions. Zick is supported by SINGA (A*STAR, Singapore) graduate fellowship. Elkind is supported by National Research Foundation (Singapore) under Research Fellowship NRF2009-08.

References

[1] R.J. Aumann and J. H. Drèze. Cooperative games with coalition structures. *International Journal of Game Theory*, 3:217–237, 1974.

[2] H. Aziz, F. Brandt, and P. Harrenstein. Monotone cooperative games and their threshold versions. In *AAMAS'10*, pages 1017–1024, 2010.

[3] H. Aziz, O. Lachish, M. Paterson, and R. Savani. Power indices in spanning connectivity games. In *AAIM'09*, pages 55–67, 2009.

[4] Y. Bachrach, E. Elkind, R. Meir, D. Pasechnik, M. Zuckerman, J. Rothe, and J. S. Rosenschein. The cost of stability in coalitional games. In *SAGT'09*, pages 122–134, 2009.

[5] Y. Bachrach, R. Meir, K. Jung, and P. Kohli. Coalitional structure generation in skill games. In *AAAI'10*, pages 703–708, 2010.

[6] J. M. Bilbao. *Cooperative Games on Combinatorial Structures*. Kluwer Publishers, 2000.

[7] H.. L. Bodlaender. Discovering treewidth. In *SOFSEM'05*, pages 1–16, 2005.

[8] G. Chalkiadakis, E. Markakis, and N. R. Jennings. Coalitional stability in structured environments. In *AAMAS'12*, 2012.

[9] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science: Volume B: Formal Models and Semantics*, pages 193–242. Elsevier, Amsterdam, 1990.

[10] G. Demange. On group stability in hierarchies and networks. *Journal of Political Economy*, 112(4):754–778, 2004.

[11] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.

[12] N. R. Devanur, M. Mihail, and V. V. Vazirani. Strategyproof cost-sharing mechanisms for set cover and facility location games. *Decision Support Systems*, 39:11–22, 2005.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractibility*. W.H. Freeman and Company, 1979.

[14] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of core, kernel, and bargaining set. *Artificial Intelligence*, 175(12–13):1877–1910, 2011.

[15] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. On the complexity of the core over coalition structures. In *IJCAI'11*, pages 216–221, 2011.

[16] S. Ieong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *EC'05*, pages 193–202, 2005.

[17] K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *STOC'01*, pages 364–372, 2001.

[18] R. Meir, Y. Bachrach, and J. S. Rosenschein. Minimal subsidies in expense sharing games. In *SAGT'10*, pages 347–358, 2010.

[19] R. Meir, J. S. Rosenschein, and E. Malizia. Subsidies, stability, and restricted cooperation in coalitional games. In *IJCAI'11*, pages 301–306, 2011.

[20] R. Myerson. Graphs and cooperation in games. *Mathematics of Operations Research*, 2(3):225–229, 1977.

[21] E. Resnick, Y. Bachrach, R. Meir, and J. S. Rosenschein. The cost of stability in network flow games. In *MFCS'09*, pages 636–650, 2009.

[22] A. D. Taylor and W. S. Zwicker. *Simple games: desirability relations, trading, pseudoweightings*. Princeton University Press, 1999.

[23] A. van den Nouweland, P. Borm, and S. Tijs. Allocation rules for hypergraph communication situations. *International Journal of Game Theory*, 20:255–268, 1992.

Reshef Meir, Jeffrey S. Rosenschein
The Hebrew University of Jerusalem, Israel
Email: `reshef24@gmail.com`, `jeff@cs.huji.ac.il`

Yair Zick, Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
Email: `yair0001@e.ntu.edu.sg`, `eelkind@ntu.edu.sg`

Manipulating Two Stage Voting Rules

Nina Narodytska and Toby Walsh

Abstract

We study the computational complexity of computing a manipulation of a two stage voting rule. An example of a two stage voting rule is Black's procedure. The first stage of Black's procedure selects the Condorcet winner if they exist, otherwise the second stage selects the Borda winner. In general, we argue that there is no connection between the computational complexity of manipulating the two stages of such a voting rule and that of the whole. However, we also demonstrate that we can increase the complexity of even a very simple base rule by adding a stage to the front of the base rule. In particular, whilst Plurality is polynomial to manipulate, we show that the two stage rule that selects the Condorcet winner if they exist and otherwise computes the Plurality winner is NP-hard to manipulate with 3 or more candidates, weighted votes and a coalition of manipulators. In fact, with any scoring rule, computing a coalition manipulation of the two stage rule that selects the Condorcet winner if they exist and otherwise applies the scoring rule is NP-hard with 3 or more candidates and weighted votes. It follows that computing a coalition manipulation of Black's procedure is NP-hard with weighted votes. With unweighted votes, we prove that the complexity of manipulating Black's procedure is inherited from the Borda rule that it includes. More specifically, a single manipulator can compute a manipulation of Black's procedure in polynomial time, but computing a manipulation is NP-hard for two manipulators.

1 Introduction

There exist several voting procedures that work in stages. For example, Black's procedure is a two stage voting rule whose first stage elects the Condorcet winner, if one exists, and otherwise moves to a second stage which elects the Borda winner [12]. As a second example, the French presidential elections use a two stage runoff voting system. If there is a majority winner in the first stage, then this candidate is the overall winner, otherwise we go to the second stage where there is a runoff vote between the two candidates with the most votes in the first round. Such two stage voting rules can inherit a number of attractive axiomatic properties from their parts. For example, Black's procedure inherits Condorcet consistency from its first part, and properties like monotonicity, participation and the Condorcet loser property from its second part. Inheriting such properties from its parts might be considered an attractive feature of two stage voting rules. On the other hand, a less desirable property of one of the base rules can infect the overall two stage rule. For instance, it has been shown that, with single peaked votes, many types of control and manipulation problems are polynomial for Black's procedure [4]. This polynomiality is essentially inherited from the first stage of the rule which selects the Condorcet winner (which must exist with single peaked votes). Such vulnerability to manipulation and control might be considered an undesirable property for a two stage voting rule. This raises several interesting questions from the perspective of computational social choice. For example, with unrestricted votes as opposed to single peaked votes, are two stage voting rules more or less computationally difficult to manipulate than single stage voting rules? How does the computational complexity of manipulating a two stage voting rule depend on the computational complexity of manipulating the two rules that it composes? In this paper, we address such questions.

Our work builds upon recent research that looks at methods to combine together voting rules. In [10], we considered a recursive combinator that successively eliminates the least popular candidate(s). This captures voting rules proposed in the past like those of Nanson, Baldwin or Coombs (all described in more detail in the next section). By comparison, we consider here a sequential combinator where the first rule eliminates all but the most popular candidates and the second rule then decides between those that remain. This captures voting rules proposed in the past like Black's procedure. Perhaps closest to this work is the sequential combinator introduced in [11]. This is an intermediate position between the two extremes of eliminating the least popular and all but the most popular candidates. Elkind and Lipmaa's combinator eliminates candidates by applying some given number of rounds of the first rule before using the second rule to decide between the candidates that remain. Even more recently, we have considered a parallel combinator that combines together the opinions of two (or more) different voting rules [16]. This combinator applies both rules simultaneously and compares their results. As well as proving computational properties of existing voting rules like Black's procedure, this paper strengthens the evidence that adding multiple rounds to voting will often increase the computational resistance to manipulation.

2 Background

A *profile* is a sequence of n total orders over m candidates. A *voting rule* is a function mapping a profile onto a set of *winners* (strictly speaking this is a social choice correspondence). We consider some of the most common voting rules.

Scoring rules: Given a *scoring vector* $(w_1, \ldots, w_m)$ of weights, the ith candidate in a vote scores w_i, and the winner is the candidate with highest total score over all the votes. The **Plurality** rule has the weight vector $(1, 0, \ldots, 0)$, the **Veto** rule has the vector $(1, 1, \ldots, 1, 0)$, and the **Borda** rule has the vector $(m-1, m-2, \ldots, 0)$.

Cup: The winner is the result of a series of pairwise majority elections between candidates. Given the *agenda*, a binary tree in which the roots are labelled with candidates, we label the parent of two nodes by the winner of the pairwise majority election between the two children. The winner is the label of the root.

Black's procedure: This rule has two stages. We first determine if there is a *Condorcet winner*, a candidate that beats all others in pairwise majority comparisons. If there is, this is the winner. Otherwise, we return the result of the *Borda* rule.

Single Transferable Vote (STV): This rule requires up to $m-1$ rounds. In each round, the candidate with the least number of voters ranking them first is eliminated until one of the remaining candidates has a majority.

Nanson's and Baldwin's rules: These are iterated versions of the Borda rule. In Nanson's rule, we compute the Borda scores and eliminate any candidate with less than half the mean score. We repeat until there is an unique winner. In Baldwin's rule, we compute the Borda scores and eliminate the candidate with the lowest score. We again repeat until there is an unique winner.

Coombs' rule: This is an iterated version of the Veto rule. We repeatedly eliminate the candidate with the most vetoes until we have one candidate with a majority.

We consider both unweighted and integer weighted votes. A weighted votes can simply be viewed as a block of identical unweighted votes.

3 Two stage voting rules

We consider a general class of two stage voting rules. Given voting rules X and Y, the rule XThen$Y$ applies the voting Y to the profile constructed by eliminating all but the winning candidates from the voting rule X. Both X and Y can themselves be two stage voting rules giving us the possibility to construct multi-stage voting rules. For example, Black's procedure is *Condorcet*THEN*Borda* where *Condorcet* is the multi-winner rule that elects the Condorcet winner if it exists, and otherwise elects all candidates. As a second example, Plurality with Runoff is *TopTwo*THEN*Majority* where *TopTwo* is the multi-winner voting rule that elects the candidates with the two most plurality votes. There are many possible rules that we might choose to combine this way. *Condorcet* is an attractive choice for the first rule as it guarantees that the resulting combination is Condorcet consistent. However, there are other interesting choices including:

CondorcetLoser: This is the rule that elects all candidates except, when it exists, the Condorcet loser.

CopelandSet: This is the rule that elects all candidates in the Copeland set. The Copeland score of a candidate is the number of candidates that it beats less the number of candidates that beats it. The Copeland set contains those candidates with the maximal Copeland score. When there is a Condorcet winner, this is the only candidate in the Copeland set.

SmithSet: This is the rule that elects all candidates in the Smith set. This is the smallest non-empty set of candidates such that every candidate in the set beats every candidate outside the set in pairwise elections. When there is a Condorcet winner, this is the only candidate in the Smith set. Voting rules like Nanson's and Kemeny are guaranteed to pick candidates from the Smith set.

SchwartzSet: This is the rule that elects all candidates in the Schwartz set. The Schwartz set is a subset of the Smith set and is the union of all the undominated sets. A set is undominated if every candidate inside the set is pairwise unbeaten by every candidate outside, and no non-empty proper subset satisfies this property. When there is a Condorcet winner, this is the only candidate in the Schwartz set.

We can also consider recursive definitions. We suppose any recursion terminates when either we have a single candidate left, or the set of candidates left does not reduce in size. For example, we can recursively define STV by *STV* = *PluralityLoser*THEN*STV* where *PluralityLoser* is the rule that elects all candidates but the candidate with the fewest first place votes. As a second example, we can recursively define Baldwin's rule by *Baldwin* = *BordaLoser*THEN*Baldwin* where *BordaLoser* is the multi-winner rule that elects all candidates but the candidate with the lowest Borda score. Nanson's rule can be defined recursively in a similar way. As a third example, we can define Coombs' rule by *Coombs* = *Majority*THEN(*VetoLoser*THEN*Coombs*) where *Majority* elects the candidate with a majority of first place votes or, if there is no such candidate, elects all candidates, and *VetoLoser* is the rule that elects all candidates but the candidate with the most last placed votes.

4 Axiomatic and algebraic properties

It is interesting to consider which axiomatic properties are inherited from the base rules being combined. For example, it is simple to see that we can inherit Condorcet consistency or the Condorcet loser properties.

Proposition 1. *For any voting rule X, the combinations Condorcet*THEN*X, CopelandSet*THEN*X, SmithSet*THEN*X and SchwartzSet*THEN*X are Condorcet consistent. Similarly, for any voting rule Y, the combination CondorcerLoser*THEN*Y satisfies the Condorcet loser property.*

With recursively defined rules, we can give a similar result. We say that a multi-winner rule is Condorcet consistent if it includes the Condorcet winner in the set of winners, and satisfies the Condorcet loser property if the set of winners never includes the Condorcet loser.

Proposition 2. *Suppose Y is recursively defined by $Y = X\text{THEN}Y$ and X is Condorcet consistent. Then Y is also Condorcet consistent. Similarly, if X satisfies the Condorcet loser property then Y does also.*

Note that the Borda loser is never the Condorcet winner. Hence, the multi-winner rule *BordaLoser* is Condorcet consistent. Thus, it follows from Proposition 2 that Baldwin's rule (which is recursively defined using *BordaLoser*) is also Condorcet consistent.

There are also axiomatic properties which can be lost by combining together voting rules. For example, the Borda loser rule which eliminates the lowest Borda scoring candidate is monotonic since increasing one's preference for a candidate can only prevent them from being the Borda loser. However, Baldwin's rule, which is the recursive version of the Borda loser rule, is not monotonic. It will therefore be interesting to identify conditions under which two stage voting rules are monotonic.

This combinator has a number of interesting algebraic properties. For example, the *Identity* rule that returns all candidates is a left and right identity of the THEN combinator. Note that the THEN combinator is neither commutative nor associative. If a voting rule is recursively defined then it is idempotent (that is, $X\text{THEN}X = X$). More complex algebraic identities can be derived such as the following.

Proposition 3. *If X is idempotent then $X\text{THEN}(X\text{THEN}Y) = X\text{THEN}Y$ and $(Y\text{THEN}X)\text{THEN}X = Y\text{THEN}X$.*

More specialized properties can also be derived such as the following.

Proposition 4. *$SmithSet\text{THEN}Nanson = Nanson$.*

Proposition 5. *If X is Condorcet consistent and only returns the Condorcet winner when they exist then $Condorcet\text{THEN}X = X$.*

5 Complexity of manipulation

One of the main contributions of this paper is to consider the impact of two stage voting rules on the computational complexity of computing a manipulation. As in previous studies (e.g. [2, 6]), we consider manipulation with unweighted votes and a small number of manipulators, and manipulation with weighted votes, a coalition of manipulators and a small number of candidates. As is common in the literature, we break ties in favour of the manipulators.

5.1 Weighted votes, general results

With weighted votes, we first argue that is no connection in general between the computational complexity of computing a manipulation of a two stage voting rule and the computational complexity of manipulating its parts.

Proposition 6. *There exist voting rules X and Y with the following properties for weighted votes:*

1. *computing coalition manipulations of X, Y and $X\text{THEN}Y$ are polynomial;*
2. *computing coalition manipulations of X and Y are polynomial but of $X\text{THEN}Y$ is NP-hard;*

3. *computing a coalition manipulation of X is polynomial and of Y is NP-hard, but of X*Then*Y is polynomial;*

4. *computing a coalition manipulation of X is polynomial, but of Y and X*Then*Y are NP-hard;*

5. *computing a coalition manipulation of X is NP-hard, but of Y and X*Then*Y are polynomial;*

6. *computing a coalition manipulation of X is NP-hard and of Y is polynomial, but of X*Then*Y is NP-hard;*

7. *computing coalition manipulations of X and Y are NP-hard but of X*Then*Y is polynomial;*

8. *computing coalition manipulations of X, Y and X*Then*Y are NP-hard.*

Proof: The NP-hardness results are derived from the NP-hardness of computing a coalition manipulation of STV with 3 or more candidates [7].

1. Consider $X = FirstRoundCup$ and $Y = Cup$. $FirstRoundCup$ is the multi-winner rule that runs one round of the Cup voting rule. Note that $FirstRoundCup$ThenCup is the Cup rule itself, and both $FirstRoundCup$ and Cup are polynomial to manipulate by a coalition even with weighted votes [7].

2. Consider $X = TopTwo$ and $Y = Majority$ where $TopTwo$ elects the two candidates with the two highest plurality scores. On 3 candidates, $TopTwo$Then$Majority$ is Plurality with runoff, which itself is equivalent STV which is NP-hard to manipulate by a coalition of weighted voters when we have 3 or more candidates [7].

3. Consider $X = Plurality'$ and $Y = STV$ where $Plurality'$ is the decisive form of plurality that includes tie-breaking in some fixed order. Note that XThen$Y$ is again $Plurality'$ which is polynomial to manipulate by a coalition even with weighted votes [7].

4. Consider $X = Identity$ and $Y = STV$ where $Identity$ is the identity rule that elects all the candidates in the election. Note that XThen$Y$ is also STV.

5. Consider $X = STV_1$ which is the multi-winner voting rule that elects both the STV winner and the candidate with the lexicographically smallest label, and Y elects the candidate with the lexicographically smallest label. Note that XThen$Y$ always elects the candidate with the lexicographically smallest label. Such a rule is polynomial to manipulate by a coalition even with weighted votes.

6. Consider $X = STV$ and $Y = Identity$. Note that XThen$Y$ is again STV.

7. Consider $X = STV_2$ and $Y = STV_3$ where STV_2 is the multi-winner rule that elects the STV winner as well as those candidates with the lexicographically smallest and largest names, and STV_3 elects the plurality winner between the candidates with the lexicographically smallest and largest names if there are 3 or fewer candidates and otherwise elects the STV winner. Note that XThen$Y$ elects the plurality winner between the candidates with the lexicographically smallest and largest names, and computing a coalition manipulation of such a rule is polynomial even with weighted votes.

8. Consider $X = Y = STV'$ where STV' is the decisive form of STV where we tie-break in favour of the manipulators. Note that XThen$Y$ is also STV'.

♡

5.2 Weighted votes, specific rules

With weighted votes, we already know that several multi-stage voting rules are NP-hard to manipulate including STV, Plurality with runoff, Baldwin's rule (all with 3 candidates), and Nanson's rule (with 4 candidates) [7, 15]. We first show that computing a manipulation of *Condorcet*THENX with weighted votes is NP-hard for any scoring rule X. This contrasts to scoring rules in general where computing a coalition manipulation is NP-hard for any rule that is not isomorphic to Plurality, but is polynomial for Plurality. This demonstrates that adding the test for a Condorcet winner to give *Condorcet*THENX increases the computational complexity of manipulation over that for the scoring rule X alone.

Proposition 7. *Deciding whether there exists a coalitional manipulation for Condorcet*THEN*Plurality with weighted votes is NP-complete with 3 or more candidates.*

Proof: We reduce from the number partitioning problem with n numbers k_i, $i = 1, \dots, n$, $\sum_{i=1}^{n} k_i = 2K$. We have n manipulators with the weight k_i each.

Consider a non-manipulator profile. Suppose voters with total weight $2K$ cast (a, b, p) and voters with total weight $2K$ cast (b, a, p). The candidate p is a Condorcet loser as it loses to both a and b. Moreover, as a and b are tied, there is no Condorcet winner.

Note that if all manipulators put p in the first position then p wins under plurality. However, the manipulators have to make sure that they also do not make a or b the Condorcet winner. Note that if a (b) gets a higher score than b (a) then a (b) is the Condorcet winner. Therefore, the only way to prevent one of them from becoming the Condorcet winner is to partition the total weight of votes between a and b. Thus, manipulators with a total weight of K have to vote (p, a, b) and the remaining manipulators have to vote (p, b, a). Therefore, there exists a manipulation iff there is a partition with the required sum K. ♡

Proposition 8. *With weighted votes and any scoring rule X that is not isomorphic to Plurality, computing a coalition manipulation of Condorcet*THENX *is NP-hard for 3 or more candidates.*

Proof: Without loss of generality, we consider a scoring rule which gives a score of α_1 for a candidate in 1st place in a vote, α_2 for 2nd place, and 0 for 3rd place. We adapt the reduction used in the proof of Theorem 6 in [8] for the NP-hardness of manipulating any scoring rule that is not isomorphic to Plurality voting. The reduction is from the number partitioning problem and constructs an election with a weight of $6\alpha_1 K - 2$ votes over the candidates a, b and p (who the manipulators wish to make win). Within these votes, the manipulators have a weight of $2(\alpha_1 + \alpha_2)K$ votes, and the rest are fixed. The number partition problem is to divide a set of integers summing to $2K$ into two equal sums. There is a manipulator of weight k_i for every integer k_i in the set being partitioned. We now add $6\alpha_1 K - 1$ triples of votes: (a, b, p), (b, p, a), (p, a, b). This has no impact on the differences in the scores between the candidates. However, it creates a Condorcet cycle so that there cannot be a Condorcet winner whatever the manipulators do with their votes. Hence, we must pass to the second round where the winner is decided by the scoring rule X. As in the proof of Theorem 6 in [8], there is a manipulation that makes p the winner of the scoring rule X iff there is a partition into two equal sums. Thus, computing a coalition manipulation of *Condorcet*THENX is NP-hard. ♡

It follows immediately that coalition manipulation of Black's procedure, which is *Condorcet*THEN*Borda* is NP-hard with 3 or more candidates.

Corollary 1. *With weighted votes, coalition manipulation of Black's procedure is NP-hard with 3 or more candidates.*

5.3 Unweighted votes, general results

As with weighted votes, there is no connection in general between the computational complexity of computing a manipulation of a two stage voting rule with unweighted votes and the computational complexity of computing a manipulation of its parts.

Proposition 9. *There exist voting rules X and Y with the following properties:*

1. *computing manipulations of X, Y and XTHENY are polynomial;*
2. *computing manipulations of X and Y are polynomial but of XTHENY is NP-hard;*
3. *computing a manipulation of X is polynomial and of Y is NP-hard, but of XTHENY is polynomial;*
4. *computing a manipulation of X is polynomial, but of Y and XTHENY are NP-hard;*
5. *computing a manipulation of X is NP-hard, but of Y and XTHENY are polynomial;*
6. *computing a manipulation of X is NP-hard and of Y is polynomial, but of XTHENY is NP-hard;*
7. *computing manipulations of X and Y are NP-hard but of XTHENY is polynomial;*
8. *computing manipulations of X, Y and XTHENY are NP-hard.*

Proof: The NP-hardness results are derived from the NP-hardness of manipulating STV with unweighted votes and a single manipulator [2].

1 Identical examples to the weighted case.

2 Consider the multi-winner voting rule X that eliminates the incumbent candidate, and the rule Y that elects the plurality winner between the candidates that are preferred by at least one voter to the incumbent or, if there are no such candidates, the STV winner. Now X is polynomial to manipulate as it ignores the votes. Similarly, Y is polynomial to manipulate since the manipulators should always put the candidate that they wish to win in first place, and the incumbent anywhere else in their vote. If all other voters prefer the incumbent to any other candidate, then this vote will ensure that the manipulators' preferred candidate wins. On the other hand, if the other voters prefer one ore more candidates to the incumbent, then this is the best vote for ensuring the manipulators' preferred candidate is the plurality winner. Now XTHENY is NP-hard to manipulate. We adapt the reduction used in [2] to prove that STV is NP-hard to manipulate by a single manipulator. We simply introduce an additional candidate, the incumbent into the voting profile used in this proof.

3-8 Identical examples to the weighted case.

♡

5.4 Unweighted votes, specific rules

With unweighted votes, we already know that a number of specific multi-stage voting rules are NP-hard to manipulate including STV [2], Nanson's, Baldwin's [15] and Coombs rules [10] (all with a single manipulator). We can add to this list Black's procedure. Like Borda voting on which it is based, a single manipulator can compute a manipulation of Black's procedure in polynomial time, but coordinating two manipulators makes the problem NP-hard.

Proposition 10. *Manipulation of Black's procedure with unweighted votes and two manipulators is NP-hard.*

Proof: We adapt the reduction used in the proof of Theorem 3.1 in [3] for the NP-hardness of manipulating Borda voting. This reduction is from a special case of numerical matching with target sums. It constructs an election with 5 votes, 3 fixed votes and 2 votes of the manipulators over the candidates 1 to m. We now add 6 sets of cyclic votes: $(1, 2, \ldots, m-1, m)$, $(2, 3 \ldots, m, 1)$, $\ldots$, $(m-1, m, \ldots, m-3, m-2)$, $(m, 1, \ldots, m-2, m-1)$. This has no impact on the differences in the scores between the candidates. However, it creates a Condorcet cycle so that there cannot be a Condorcet winner whatever the manipulators do with their two votes. Hence, we must pass to the second round where the winner is decided by the Borda rule. As in the proof of Theorem 3.1 in [17], there is a manipulation that makes a chosen candidate the Borda winner iff there is a solution to the numerical matching problem with target sums. Thus, computing a manipulation of *Condorcet*THEN*Borda*, which is Black's procedure, is NP-hard. ♡

Proposition 11. *Deciding whether one manipulator can make a candidate win for Black's procedure with unweighted votes is polynomial.*

Proof: We consider several cases.

Suppose no Condorcet winner exists in the profile P of votes of the non-manipulators, but there are $a \neq p$ and $b \neq p$ such that $beat_P(a, b) = beat_P(b, a)$, where $beat_P(v_1, v_2)$ is the number of times v_1 beats v_2 in P. In this case, p loses regardless of how the manipulator votes as the manipulator's vote must give an advantage of one vote to a or b. Hence, one of a or b must be the Condorcet winner.

Suppose no Condorcet winner exists in P and there is **no** $a \neq p$ and $b \neq p$ such that $beat_P(a, b) = beat_P(b, a)$. Then the manipulator casts a vote using to the greedy rule. This vote does not create a Condorcet winner that is different from p, hence it is optimum for both the Condorcet criterion and Borda rule.

Suppose there is a Condorcet winner in P, $a \neq p$. If there is no b such that $beat_P(a, b) = beat_P(b, a) + 1$ then a is the winner regardless of the manipulator's vote. Therefore, suppose there exists a set B such that $beat_P(a, b) = beat_P(b, a) + 1$, $b \in B$. If there exists b such that $score_P(a) \geq score_P(b)$ then a will be ranked below b in the manipulator's vote that is constructed based on the greedy algorithm (or we can swap a and b if their scores are equal). Therefore,we assume that $score_P(a) < score_P(b)$. Let b^* be the candidate with the minimum score $score_P$, so that $b^* = argmin_{b \in B}(score_P(b))$. The manipulator must rank a below b^* to prevent a from being the Condorcet winner. This is equivalent to assuming that $score_P(a) = score_P(b^*)$ and using the greedy algorithm to construct the manipulator's vote. If this is a successful manipulation then we are done. If it is not then there is no way to construct a successful manipulation. ♡

6 Multiple ballots

So far, we have assumed that voters vote only once. However, the THEN combinator is naturally sequential. We can therefore consider the case where voters are allowed to re-vote in each round. For example, in the French presidential elections, voters re-vote in the second stage. Such re-voting increases the potential for manipulation in two ways. First, as we illustrate here, there are elections which can only be manipulated when the manipulators vote differently in the two rounds. Of course, all those elections where manipulators can change the result by strategically voting the same way in both rounds remain manipulable. Second, as we also argue in the next section, the first round of voting reveals voters' preferences, thereby enabling manipulations to take place that require such knowledge. Third, voters can vote strategically in the first round to give their preferred candidate an easier contest in the second round.

If voters re-vote between rounds, we add "with re-voting" to its name. Hence, plurality with runoff and re-voting is the two stage election rule used in French presidential elections in which, unless there a majority in the first round, plurality is used in the first round to select two candidates to go through to the runoff, and voters then re-vote in the second round to decide the winner of the runoff. The following example demonstrates that there exist elections where strategic voting with plurality with runoff is not possible, but is with plurality with runoff and re-voting.

Example 1. *Suppose we have 2 votes for* (a, b, p)*, 2 votes for* (b, a, p)*, 1 vote for* (b, p, a)*, 2 votes for* (p, a, b) *and 2 manipulators whose preferences are* (p, a, b)*. In addition, we suppose in the event of a tie in the first round between all 3 candidates, the manipulators' preferred candidate* p *and* a *go through to the runoff. Note that if the manipulators vote truthfully, then* p *and* b *have the most votes in the first round, and* b *wins the runoff by 5-4. To make* p *the winner, the manipulators need* a *and* p *to be in the runoff. This is possible if and only if one of the manipulators votes for* a *and the other votes for* p *in the first round. We then have a 3-way tie and, according to the tie-breaking rule,* a *and* p *go through to the runoff. If the manipulators do not re-vote in the runoff,* a *wins the runoff by 5-4. On the other hand, if the manipulators can re-vote in the runoff, both can vote for* p*, and* p *will beat* a *by 5-4.*

7 Revealed preferences

One of the strong assumptions made in much work on (the complexity of) manipulation is that the manipulators know the other voters' preferences [9]. There are many situations where this is unrealistic. When we have re-voting, it is reasonable to suppose voters' preferences have been (partially) revealed by the first round of voting. This introduces new opportunities for manipulation. Consider Black's procedure with re-voting and a manipulator who lacks any knowledge of the other voters' preferences, so votes truthfully in the first round. The following example demonstrates that this manipulator can vote strategically in the second round based on the votes revealed in the first round.

Example 2. *Suppose the first round reveals that there are 2 votes for* (a, b, p)*, 2 votes for* (b, p, a)*, 1 vote for* (p, a, b)*, and a single manipulator's truthful vote for* (p, b, a)*. There is no Condorcet winner so all candidates go through to the second round. Without re-voting,* b *has the highest Borda score in the second round and is the overall winner. On the other hand, suppose the manipulator changes their vote in the second round to* (p, a, b) *based on the preferences revealed in the first round. Then, assuming the other votes remain the same, the Borda scores of all candidates are equal. If such a 3-way tie is broken in favour of the manipulator, then the manipulator's preferred candidate* p *now wins.*

It is natural to consider more game theoretic behaviours in such two stage voting rules. Re-voting can be viewed as a finite repeated sequential game so we can use concepts like subgame perfect Nash equilibrium and backward induction to predict how agents will play strategically in each round. An interesting open question is the computational complexity of computing such strategic behaviour. This sort of strategic voting has already received some attention in the literature. For example, Bag, Sabourian and Winter prove that a class of voting rules which use repeated ballots and eliminate one candidate in each round are Condorcet consistent [1]. They illustrate this class with the *weakest link* rule in which the candidate with the fewest ballots in each round is eliminated.

It is also natural to consider iterated voting in multiple stage voting rules. After each round of voting, we might suppose that agents change their vote according to a best response strategy, starting perhaps from a truthful vote. We can also consider the situation where the full preferences of the agents are revealed in each round of voting, as well as the situation where only partially information is revealed like total Borda scores. However, unlike previous studies like [14], candidates are also eliminated in each round.

8 Related work

As noted earlier, a number of well known voting rules like Black's procedure and Plurality with runoff are instances of this voting schema. However, there exist many other related voting rules. For example, Conitzer and Sandholm [5] studied the impact on the computational complexity of manipulation of adding an initial round of the Cup rule to a voting rule X. This initial round eliminates half the candidates and makes manipulation NP-hard to compute for several voting rule including plurality and Borda. Consider the multi-winner voting rule, *Bisect* which runs an election between given pairs of candidates, and returns the winning half of the candidates. Then Conitzer and Sandholm's study can be viewed as of the two stage voting rule *Bisect*THENX. Elkind and Lipmaa [11] extended this idea to a general technique for combining two voting rules. The first voting rule is run for some number of rounds to eliminate some of the candidates, before the second voting rule is applied to the candidates that remain. They proved that many such combinations of voting rules are NP-hard to manipulate.

Beside STV, Nanson's, Baldwin's and Coombs rule, a number of other recursively defined rules have been put forwards in the literature. For example, Tideman proposed the Alternative Smith rule [18]. This is recursively defined as *SmithSet*THEN(*PluralityLoser*THEN*AlternativeSmith*). Other complex multi-stage rules have also been proposed. For example, [13] has proposed a complex rule that computes the Schwartz choice set, then iteratively applies Copeland's procedure to this set until a fixed point is reached. If several candidates remain at this point, the rule then selects the plurality winners. If there are several such winners, the rule then chooses among then according to the number of second place votes, and so on. If this still does not select a winner, a lottery is then used amongst the candidates that remain.

We recently proposed a combinator for taking the consensus of two (or more) voting rules. Given two voting rules X and Y, the combinator $X + Y$ computes the winners of X and Y and then recursively applies $X + Y$ to this set. If X and Y are majority consistent (that is, given an election with just two candidates, they both return the majority winner) then $X + Y$ is (XORY)THEN$Majority$ where XORY returns the union of the winners of X and Y.

9 Conclusions

We have considered voting rules which have multiple stages. For example, Black's procedure selects the Condorcet winner if they exist, otherwise in the second stage, it selects the Borda winner. We denoted this as *Condorcet*THEN*Borda*. Combining voting rules together in this way can increase their resistance to manipulation. For example, whilst Plurality is polynomial to manipulate with weighted votes, *Condorcet*THEN*Plurality* is NP-hard with 3 or more candidates and a coalition of manipulators. A combination of voting rules can also inherit computational resistance to manipulation from its part. For example, we proved that computing a manipulation of Black's procedure, which is *Condorcet*THEN*Borda*, is NP-hard with weighted or unweighted votes. There are many directions for future work. For instance, it would also be interesting to consider the impact of such two stage voting on other types of control, on bribery and on issues like the computation of possible winners.

Acknowledgements

NICTA is funded by the Department of Broadband, Communications and the Digital Economy, and the Australian Research Council. The authors are also supported by the Asian Office of Aerospace Research and Development (AOARD).

References

[1] P.K. Bag, H. Sabourian, and E. Winter. Multi-stage voting, sequential elimination and Condorcet consistency. *Journal of Economic Theory*, 144(3):1278 – 1299, 2009.

[2] J.J. Bartholdi and J.B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[3] N. Betzler, R. Niedermeier, and G.J. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In T. Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*. International Joint Conference on Artificial Intelligence, 2011.

[4] F. Brandt, M. Brill, E. Hemaspaandra, and L.A. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In M. Fox and D. Poole, editors, *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*. AAAI Press, 2010.

[5] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of 18th IJCAI*, pages 781–788. International Joint Conference on Artificial Intelligence, 2003.

[6] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate. *Journal of the Association for Computing Machinery*, 54, 2007.

[7] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate. *Journal of the Association for Computing Machinery*, 54, 2007.

[8] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate. *Journal of the Association for Computing Machinery*, 54, 2007.

[9] V. Conitzer, T. Walsh, and L. Xia. Dominating manipulations in voting wih partial information. In W. Burgard and D. Roth, editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*. AAAI Press, 2011.

[10] J. Davies, N. Narodytska, and T. Walsh. Eliminating the weakest link: Making manipulation intractable? In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2012)*. AAAI Press, 2012.

[11] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC'05)*, 2005.

[12] P.C. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[13] C.G. Hoag and G.H. Hallett. *Proportional Representation*. Macmillan, 1926.

[14] O. Lev and J.S. Rosenschein. Convergence of iterative voting. In *The Eleventh International Joint Conference on utonomous Agents and Multiagent Systems (AAMAS 2012)*, Valencia, Spain, June 2012.

[15] N. Narodytska, T. Walsh, and L. Xia. Manipulation of Nanson's and Baldwin's rules. In W. Burgard and D. Roth, editors, *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*. AAAI Press, 2011.

[16] N. Narodytska, T. Walsh, and L. Xia. Combining voting rules together. In Luc de Raedt, editor, *Proc. of the 20th European Conference on Artificial Intelligence (ECAI-2012)*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2012.

[17] B. Reilly. Social choice in the south seas: electoral innovation and the borda count in the pacific island countries. *International Political Review*, 23(4):355–372, 2002.

[18] N. Tideman. *Collective Decisions And Voting: The Potential for Public Choice*. Ashgate, 2006.

Nina Narodytska and Toby Walsh
NICTA and UNSW
Sydney, Australia
Email: `{nina.narodytska,toby.walsh}@nicta.com.au`

Complexity and Approximability of Social Welfare Optimization in Multiagent Resource Allocation[1]

Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe

Abstract

A central task in multiagent resource allocation, which provides mechanisms to allocate (bundles of) resources to agents, is to maximize social welfare. We assume resources to be indivisible and nonshareable and agents to express their utilities over bundles of resources, where utilities can be represented in the bundle form, the *k*-additive form, and as straight-line programs. We study the computational complexity of social welfare optimization in multiagent resource allocation, where we consider utilitarian and egalitarian social welfare and social welfare by the Nash product. We prove that exact social welfare optimization by the Nash product is DP-complete for the bundle and the 3-additive form, where DP is the second level of the boolean hierarchy over NP. For utility functions represented as straight-line programs, we show NP-completeness for egalitarian social welfare optimization and social welfare optimization by the Nash product. Finally, we show that social welfare optimization by the Nash product in the 1-additive form is hard to approximate, yet we also give fully polynomial-time approximation schemes for egalitarian and Nash product social welfare optimization in the 1-additive form with a fixed number of agents.

1 Introduction

Multiagent resource allocation (MARA) deals with distributing resources to agents that have preferences over (bundles of) resources. These resources are assumed to be indivisible and nonshareable. Agents express their preferences by means of utility functions. Hence, every given allocation of resources to agents induces a vector of utilities that can be aggregated to a single value, the social welfare of this allocation. There are different notions of social welfare, ranging from the well-known utilitarian social welfare to egalitarian social welfare, to compromises between these two notions such as the Nash product and generalizations thereof (*k*-rank dictator functions, etc.).

In a bit more detail, *utilitarian social welfare* sums up the agents' individual utilities in a given allocation, thus providing a useful measure of the overall—and also of the average—benefit for society. For instance, in a combinatorial auction the auctioneer's aim is to maximize the auction's revenue (i.e., the sum of the prizes paid for the items auctioned), no matter which agent can realize which utility.

In contrast, *egalitarian social welfare* gives the utility of the agent who is worst off in a given allocation, which provides a useful measure of fairness in cases where the minimum needs of all agents are to be satisfied. For example, think of distributing humanitarian aid items (such as food, medical aid, blankets, tents, etc.) among the needy population in a disaster area (e.g., an area hit by an earthquake or a tsunami). Guaranteeing every survivor's continuing survival is the primary goal in such a scenario, and it is best captured by the notion of egalitarian social welfare.

As mentioned above, the *Nash product*, the product of the agents' utilities, can be seen as a compromise between these two approaches. On the one hand, it has the (strict) monotonicity property of utilitarian social welfare because an increase in any agent's utility leads to an increase of the Nash product (provided all agents have positive utility). On the other hand, the Nash product increases

[1]Preliminary versions of parts of this paper appear in the proceedings of the *11th International Joint Conference on Autonomous Agents and Multiagent Systems* [17], of the *6th European Starting AI Researcher Symposium* [16], and of the *12th International Symposium on Artificial Intelligence and Mathematics* [18]. This work was supported in part by DFG grant RO-1202/14-1, ARC grant DP110101792, a DAAD grant for a PPP project in the PROCOPE program, and a fellowship from the Vietnamese government.

as well when reducing inequitableness among agents by redistributing utilities, thereby providing a measure of fairness. Looking at the ordering that is induced by the allocations, the "social welfare ordering," Moulin [15] presents further beneficial properties of the Nash product. For example, the Nash product is uniquely characterized by independence of individual scale of utilities,[2] i.e., even if different "currencies" are used to measure the agents' utilities, the social welfare ordering remains unaffected.

All these notions of social welfare have in common that they seek to model that a high value of social welfare implies well-being among the society of agents (that is, for the group as a whole). Thus, the goal is to find allocations that maximize social welfare. How difficult is this task? The main purpose of this paper is to find answers to this question—for various central notions of social welfare, for distinct ways of representing utility functions, and for different ways of modeling MARA problems.

Although resource allocation problems are important for human agents as well, we are mostly concerned with (autonomous) software agents having individual utilities and acting in a shared environment (e.g., in a multiagent system). Therefore, it is of particular interest to study the computational complexity of MARA problems and to tackle computational hardness results by means of approximation algorithms. We present NP-completeness results for decision problems associated with egalitarian and Nash product social welfare optimization, and DP-completeness results for decision problems associated with Nash product social welfare optimization. We complement our results on the computational complexity of MARA decision problems by proving that the Nash product social welfare optimization problem is hard to approximate in the 1-additive form. For a fixed number of agents, we also give fully polynomial-time approximation schemes for egalitarian and of Nash product social welfare optimization in the 1-additive form.

This paper is organized as follows: In Section 2, we formalize the MARA framework that we have adopted from the profound survey by Chevaleyre et al. [5], and we introduce the needed background from complexity theory, including the perhaps lesser known complexity class DP, as well as some basic notions of approximation theory. Then, in Section 3, we briefly survey related work to see the context of our results. In Section 4 we present computational complexity results for the decision versions related to social welfare optimization, and in Section 5 we are concerned with approximability of social welfare optimization. In Section 6, we summarize our results and conclude with some open questions.

2 Preliminaries

2.1 Multiagent Resource Allocation Settings

We adopt the framework for multiagent resource allocation described in the survey by Chevaleyre et al. [5]. Let $A = \{a_1, a_2, \ldots, a_n\}$ be a set of n agents and let $R = \{r_1, r_2, \ldots, r_m\}$ be a set of m indivisible and nonshareable resources (i.e., each resource is assigned as a whole and can be assigned to only one agent). Subsets of R are called *bundles of resources*.

Every agent associates utility to every bundle of resources by specifying a utility function $u_i : 2^R \to \mathbb{F}$, where 2^R denotes the power set of R and $\mathbb{F}$ is a numerical set (such as the set $\mathbb{N}$ of nonnegative integers, the set $\mathbb{Z}$ of integers, the set $\mathbb{Q}$ of rational numbers, and the set $\mathbb{Q}^+$ of nonnegative rational numbers). The idea behind utility functions mapping *bundles* of resources rather than single resources to values in $\mathbb{F}$ is that agents might be willing to pay either more or less for a bundle than the sum of their utilities for this bundle's single items. For example, owning a pair of matching shoes is likely to be more valuable to an agent than the sum of the values each single shoe has for this agent. On the other hand, an agent who is willing to bid on 100 identical items might expect

[2]Similarly, utilitarian social welfare is characterized by independence of individual zeros of utilities: A constant shift of an agent's utility function does not change the social welfare ordering.

some discount and so has less utility for the bundle of 100 items than 100 times the utility assigned to a single item.

Let $U = \{u_1, u_2, \dots, u_n\}$ be the set of the agents' *utility functions*. A triple (A,R,U) is called a *multiagent resource allocation setting* (a *MARA setting*, for short).

A concrete distribution of resources to agents is an allocation. Formally, for a given MARA setting (A,R,U), an *allocation* is a mapping

$$X : A \to 2^R$$

with $\bigcup_{a_i \in A} X(a_i) = R$ (i.e., every resource is given to some agent) and $X(a_i) \cap X(a_j) = \emptyset$ for any two distinct agents a_i and a_j (i.e., no resources are given to multiple agents). The set of all allocations for a MARA setting (A,R,U) is denoted by $\Pi_{A,R}$ and has cardinality n^m. We use the shorthand $u_i(X)$ to denote the utility $u_i(X(a_i))$ agent a_i can realize in allocation X because we assume agents not to be interested in externalities.

2.2 Representations of Utility Functions

Utility functions can be given in different ways, and the representation form potentially affects the complexity of the corresponding problems. We consider the following representation forms for utility functions:

1. **The bundle form:** A utility function $u : 2^R \to \mathbb{F}$ is in *bundle form* if it is represented by a list of pairs $(R', u(R'))$ for any bundle $R' \subseteq R$, omitting pairs with zero utility. This representation form is "fully expressive" (i.e., every utility function can be described in bundle form), but its drawback is a potentially exponential representation size in the number of resources.

2. **The k-additive form, for some fixed positive integer k:** A utility function $u : 2^R \to \mathbb{F}$ is in *k-additive form* if for each bundle $T \subseteq R$ with $\|T\| \leq k$, there is a unique coefficient $\alpha_T \in \mathbb{F}$ such that for every bundle $R' \subseteq R$ the following holds:

   $$u(R') = \sum_{T \subseteq R', \|T\| \leq k} \alpha_T.$$

   Sometimes we write (T, ℓ) for the coefficient $\alpha_T = \ell$. This coefficient expresses the "synergetic" value of some agent owning all the resources in T. This representation form is fully expressive only if k is large enough. On the other hand, choosing k to be relatively small allows for a relatively succinct representation of utility functions. Originally, Grabisch [11] defined the k-additive form. However, in multiagent resource allocation it was proposed for representing utilities by Chevaleyre et al. [6, 7] and, independently, in combinatorial auctions by Conitzer et al. [8].

3. **Straight-line program representation:** Informally, a straight-line program is a topologically sorted list of gates of a boolean circuit C that takes as input an m-dimensional binary vector and outputs s bits. Interpreting the input vector as a bundle of resources R' and the output as the binary representation of $u(R')$, we can say that C (or a corresponding straight-line program) represents utility function u.

   Formally (see, e.g., [9]), an *(m,s)-combinational logic network* is a directed graph with m input nodes $(\beta_1, \dots, \beta_m)$ of in-degree 0, s output nodes $(\gamma_{s-1}, \dots, \gamma_0)$ of out-degree 0, and gate nodes of in-degree at most 2 and out-degree at least 1. A gate node represents one of the common boolean operations $(\wedge, \vee, \neg)$. An input to the nodes $(\beta_1, \dots, \beta_m)$ can be interpreted as a vector of length m and vice versa. Hence, every input vector β induces[3] an output vector

[3] Every bit at a gate node is induced as usual: If a is a gate node with a 2-ary boolean operation σ, then the bit induced at a is $b_1 \sigma b_2$, provided that (b_1, a) and (b_2, a) are edges of the graph, σ is a binary operation, and by b_1 and b_2 we mean the induced bits at nodes b_1 and b_2. For the boolean operation $\neg$, the definition is analogous.

$C(\beta)$, where we denote by $C(\beta)_i$ the i-th least significant bit of $C(\beta)$. Let $R = \{r_1, \dots, r_m\}$, let $u : 2^R \to \mathbb{N}$ be a utility function and C an (m,s) combinational logic network. Denote by β_S the characteristic vector that has for every $j \in \{1, \dots, m\}$ the j-th coordinate equal 1 if and only if $r_j \in S$ for some $S \subseteq R$. Utility function u is *realized* by C if for every $S \subseteq R$ with binary vector β_S the following holds:

$$u(S) = \sum_{i=0}^{s-1} 2^i \cdot C(\beta_S)_i.$$

The advantages of straight-line programs are mainly the efficiency of evaluation (linear time in the number of nodes) and its conciseness, which is supported by the following result by Pippenger and Fischer [22] and Schnorr [27].

Fact 1 *Let $f : \{0,1\}^m \to \{0,1\}^s$. If there exists a deterministic Turing machine that computes f in time T, then there exists a straight-line program of $\mathcal{O}(T \log T)$ lines that computes f as well.*

In multiagent resource allocation, utility representations by straight-line programs were introduced by Dunne et al. [9].

2.3 Measures of Social Welfare

The notion of social welfare is a tool to assess and rank allocations based on specific measures of quality. Thus, different allocations might be "the best allocation," depending on the notion of social welfare that is employed. We will study the following notions of social welfare.

Definition 2 *For a MARA setting (A,R,U) and an allocation $X \in \Pi_{A,R}$, define*

1. *the* utilitarian social welfare of X *as* $sw_u(X) = \sum_{a_i \in A} u_i(X)$;
2. *the* egalitarian social welfare of X *as* $sw_e(X) = \min_{a_i \in A}\{u_i(X)\}$;
3. *the* Nash product of X *as* $sw_N(X) = \prod_{a_i \in A} u_i(X)$.
4. *As an additional notation, for $S \in \{u,e,N\}$, denote the* maximum utilitarian/egalitarian/ Nash product social welfare of a MARA setting $M = (A,R,U)$ *(or of a problem instance that contains a MARA setting M) by*

   $$\max_S(M) = \max\{sw_S(X) \mid X \in \Pi_{A,R}\}.$$

   We write $\max(M)$ *for* $\max_S(M)$ *when S is clear from context.*

2.4 Problems Modeling Social Welfare Optimization

We are now ready to formally define the problems modeling social welfare optimization in multiagent resource allocation. We start with the decision problems. For $\mathbb{F} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}^+, \mathbb{Q}\}$ and form $\in \{\text{bundle}\} \cup \{k\text{-add} \mid k \geq 1\} \cup \{\text{SLP}\}$, where k-add abbreviates "k-additive" and SLP "straight-line program," define:

$\mathbb{F}$-NASH PRODUCT SOCIAL WELFARE OPTIMIZATION$_{\text{form}}$	
Given:	A MARA setting $M = (A,R,U)$, where form indicates how every $u_i : 2^R \to \mathbb{F}$ in U is represented, and a threshold $t \in \mathbb{F}$.
Question:	Is there an allocation $X \in \Pi_{A,R}$ such that $sw_N(X) \geq t$?

We abbreviate this problem by $\mathbb{F}$-NPSWO$_{\text{form}}$ (sometimes omitting the prefix "$\mathbb{F}$-"). The exact version of this problem is denoted by $\mathbb{F}$-EXACT NASH PRODUCT SOCIAL WELFARE OPTIMIZATION$_{\text{form}}$ (or, for short, by $\mathbb{F}$-XNPSWO$_{\text{form}}$) and asks, given a MARA setting $M = (A,R,U)$ and a target $t \in \mathbb{F}$, whether $\max_N(M) = t$.

The corresponding problems for utilitarian and egalitarian social welfare can be defined analogously and are abbreviated by $\mathbb{F}$-USWO$_{\text{form}}$ and $\mathbb{F}$-ESWO$_{\text{form}}$, respectively.

Apart from decision problems we also consider the corresponding three maximization problems, one for each type of social welfare. For example, the maximization problem for utilitarian social welfare is formally defined as follows:

$\mathbb{F}$-MAXIMUM UTILITARIAN SOCIAL WELFARE$_{\text{form}}$	
Input:	A MARA setting $M = (A,R,U)$, where form indicates how every $u_i : 2^R \to \mathbb{F}$ in U is represented.
Output:	$\max_u(M)$.

As a shorthand, write $\mathbb{F}$-MAX-USW$_{\text{form}}$. Based on sw_e and sw_N, $\mathbb{F}$-MAXIMUM EGALITARIAN SOCIAL WELFARE$_{\text{form}}$ (or $\mathbb{F}$-MAX-ESW$_{\text{form}}$) and $\mathbb{F}$-MAXIMUM NASH PRODUCT SOCIAL WELFARE$_{\text{form}}$ (or $\mathbb{F}$-MAX-NPSW$_{\text{form}}$) are defined accordingly.

2.5 Some Background on Complexity Theory and Approximation Theory

We assume basic knowledge of complexity theory, in particular of the complexity classes P, NP, and coNP, of central notions such as (polynomial-time many-one) reducibility (denoted by $\leq_m^p$), hardness and completeness of a problem for a complexity class with respect to $\leq_m^p$, etc. (see, e.g., the textbooks by Garey and Johnson [10], Papadimitriou [20], and Rothe [25]).

Papadimitriou and Yannakakis [21] introduced the complexity class DP, which consists of the differences of any two NP-problems. DP is the second level of the boolean hierarchy over NP and it is widely assumed that NP and coNP are both strictly contained in DP.

Typical DP problems are UNIQUE SATISFIABILITY ("*Does a given boolean formula have exactly one satisfying assignment?*") and exact variants of optimization problems such as the exact version of the TRAVELING SALESPERSON PROBLEM (EXACT-TSP): "*Given a graph and an integer t, does a shortest traveling salesperson tour have length exactly t?*" Intuitively, this problem is potentially harder than the usual TSP because both an NP problem ("*Does there exist a tour of length at most t, i.e., is the minimum tour length at most t?*," which *is* the usual TSP) and a coNP problem ("*Do all tours have length at least t, i.e., is the minimum tour length at least t?*") have to be solved to solve EXACT-TSP, which is complete for DP.

Turning to approximation theory, we define approximation algorithms for maximization problems and polynomial-time approximation schemes. Then we discuss reducibilities to prove inapproximability results.

Definition 3 (α-approximation algorithm) *Let Π be a maximization problem and $\alpha : \mathbb{N} \to (0,1)$. An α-*approximation algorithm *A for Π is a polynomial-time algorithm such that for each instance x of Π,*

$$val(A(x)) \geq \alpha(|x|) \cdot OPT(x),$$

where $val(A(x))$ denotes the value of a solution produced by A on input x and where $OPT(x)$ denotes the value of an optimal solution for x.

The *approximation factor* α might be a constant function such as $1-\varepsilon$ for some ε, $0 < \varepsilon < 1$, or a function of the input size, such as $1/\log n$ and $1/n^c$ for some $c > 0$.

Definition 4 (FPTAS) *A maximization problem* Π *has a* fully polynomial-time approximation scheme *(*FPTAS*) if for each* ε*,* $0 < \varepsilon < 1$*, there exists a* $(1-\varepsilon)$*-approximation algorithm* A_ε *for* Π*, where the running time is polynomial in* $1/\varepsilon$ *as well.*

One approach to prove inapproximability for a maximization problem is to find an α-gap-introducing reduction from an NP-complete problem.

Definition 5 (α-gap-introducing reduction) *Let* $A \subseteq \Sigma^*$ *be an* NP*-complete problem,* Π *be a maximization problem, and let* $\alpha : \mathbb{N} \to [0,1]$ *be a polynomial-time computable function of the input size. An* α-gap-introducing reduction from A to Π *is given by two polynomial-time computable functions* f *and* g *such that for each* $x \in \Sigma^*$*,*

1. $g(x)$ *is an instance of* Π*,*
2. *if* $x \in A$ *then* $OPT(g(x)) \geq f(x)$*, and*
3. *if* $x \notin A$ *then* $OPT(g(x)) < \alpha(|x|) \cdot f(x)$*.*

Note that an α-approximation algorithm B for a maximization problem Π that has an α-gap-introducing reduction from an NP-complete problem A implies $x \in A$ if and only if the value of the solution $B(g(x))$ is at least $\alpha(|x|) \cdot f(x)$. Hence, there can be no α-approximation algorithm for Π, unless P = NP.

Definition 6 (L-reduction) *Let* Π_1 *and* Π_2 *be some maximization problems. An* L-reduction from Π_1 to Π_2 *is given by two polynomial-time computable functions* f *and* g *and two parameters* α *and* β *such that for each instance* x *of* Π_1*,*

1. $y = f(x)$ *is an instance of* Π_2*,*
2. $OPT(y) \leq \alpha \cdot OPT(x)$*, and*
3. *for each solution* s_2 *for* y *of value* v_2*,* $s_1 = g(s_2)$ *is a solution for* x *of value* v_1 *such that*

$$OPT(x) - v_1 \leq \beta \cdot (OPT(y) - v_2).$$

Having an L-reduction from maximization problem Π_1 to Π_2 with parameters α,β and an $(1-\varepsilon)$-approximation algorithm for Π_2 implies a $(1-\alpha\beta\varepsilon)$-approximation algorithm for Π_1 by invoking f on the instance x of Π_1 to get an instance y of Π_2, then running the approximation algorithm for Π_2 on y and, at last, translating the solution back via g. Note that if Π_1 does not admit a $(1-\varepsilon)$-approximation algorithm and reduces to Π_2 with parameters $\alpha = \beta = 1$ then Π_2 cannot have a $(1-\varepsilon)$-approximation algorithm either.

For more background on approximation theory, see, e.g., the textbook by Vazirani [28] and the survey by Arora and Lund [1].

3 Related Work

The first paper concerned with classifying MARA problems in terms of their complexity is due to Chevaleyre et al. [6], see also [7]. They showed that the decision problem associated with utilitarian social welfare optimization is NP-complete for both the bundle and the k-additive form. Dunne et al. [9] proved that the problem remains NP-complete if utility functions are represented by straight-line programs. For further results on the complexity of fair allocation problems, we refer to Bouveret's thesis [3].

Roos and Rothe [24] proved NP-completeness for egalitarian social welfare optimization and social welfare optimization by the Nash product for the bundle form and for the k-additive form.

Table 1: Complexity of decision problems for (exact) social welfare optimization. Key: NP-c means "NP-complete" and DP-c means "DP-complete."

Social Welfare	Bundle	Reference	k-Additive	Reference
Utilitarian	NP-c	Chevaleyre et al. [7]	NP-c, $k \geq 2$	Chevaleyre et al. [7]
Egalitarian	NP-c	Roos & Rothe [24]	NP-c, $k \geq 1$	Roos & Rothe [24] and Lipton et al. [14]
Nash Product	NP-c	Roos & Rothe [24] and Ramezani & Endriss [23]	NP-c, $k \geq 1$	Roos & Rothe [24]
Exact Utilitarian Exact Egalitarian	DP-c	Roos & Rothe [24]	DP-c, $k \geq 2$	Roos & Rothe [24]
Exact Nash Product	DP-c	Theorem 7	DP-c, $k \geq 3$	Theorem 8

Social Welfare	SLP	Reference
Utilitarian	NP-c	Dunne et al. [9]
Egalitarian Nash Product	NP-c	Theorem 11

In addition, they proved DP-completeness for exact utilitarian and exact egalitarian social welfare optimization for both representation forms. Lipton et al. [14] provided a reduction to prove NP-hardness of finding a minimum-envy allocation (i.e., an allocation X that minimizes the envy $\max_{i,j}\{0, u_i(X(a_j)) - u_i(X(a_i))\}$). This reduction proves NP-hardness of the decision problem associated with egalitarian social welfare optimization as well. Independently of the result of Roos and Rothe [24], Ramezani and Endriss [23] proved the same NP-completeness result of Nash product social welfare optimization for the bundle form. Previous completeness results are summed up together with our results in Table 1.

Known approximability and inapproximability results in multiagent resource allocation have been surveyed recently in [19].

4 Complexity of Decision Problems Associated with Social Welfare Optimization

4.1 Utilities in the Bundle Form and the k-Additive Form

Roos and Rothe [24] conjectured that exact social welfare optimization by the Nash product is DP-complete for the bundle form and for the k-additive form. We confirm their conjecture in the affirmative.

It might be tempting to think that hardness for the decision problem associated with utilitarian social welfare optimization directly transfers to that for the Nash product by the straightforward reduction that maps utilities of value k to 2^k (cf. [23]). Note that not the exponential blow-up of the numbers encoding utilities causes a problem here, since the reduction from SET PACKING that Chevaleyre et al. [7] define to show NP-hardness of $\mathbb{Q}$-USWO$_{\text{bundle}}$ yield instances with utilities zero or one only. However, the reason for why this reduction doesn't work for the bundle form is that utilities of value zero that are omitted in the instances for utilitarian social welfare need to be encoded by the value $2^0 = 1$ in the resulting instance for the Nash product. In the worst case, this causes an exponential increase in the size of the instance constructed, and thus the reduction is not polynomial-time.

Relatedly, another reason for why Nash product and utilitarian social welfare are not equivalent is that if we make the plausible assumption that the empty bundle has utility zero for everyone, the Nash product is trivially zero when there are fewer resources than agents (which implies that at least one agent must remain empty-handed), while utilitarian social welfare is not in that case.

Theorem 7 $\mathbb{Q}^{+}$-XNPSWO$_{\text{bundle}}$ *is* DP*-complete.*

Theorem 8 *For each $k \geq 3$,* $\mathbb{Q}^{+}$-XNPSWO$_{k\text{-add}}$ *is* DP*-complete.*

The proofs of Theorems 7 and 8 are omitted due to space limitations. In order to prove DP-hardness for $\mathbb{Q}^{+}$-XNPSWO$_{\text{bundle}}$, we need the following lemma by Chang and Kadin [4], who provided a sufficient condition for DP-hardness. It makes use of the definition of AND_2.

Definition 9 *Let $L \subseteq \Sigma^*$ be a decision problem. L has AND_2 if there exists a polynomial-time computable function f such that for all strings $x, y \in \Sigma^*$, it holds that*

$$x \in L \wedge y \in L \iff f(x,y) \in L.$$

Lemma 10 (Chang and Kadin [4]) *Let $L \subseteq \Sigma^*$ be a decision problem. If L is both* NP*-hard and* coNP*-hard and has AND_2, then L is* DP*-hard.*

We roughly present the idea of the proofs of Theorems 7 and 8. First, note that the proof of NP-hardness of $\mathbb{Q}^{+}$-XNPSWO$_{\text{bundle}}$ (see [24]) in fact proves coNP-hardness of this problem as well, and this reduction produces MARA settings, where the agents' utility functions take on binary values only. Since this is a special case of $\mathbb{Q}^{+}$-XNPSWO$_{\text{bundle}}$, hardness results carry over. To apply Lemma 10, it remains to show that any two instances can be merged in the sense of AND_2. Note that trivial merging of two $\mathbb{Q}^{+}$-XNPSWO$_{\text{bundle}}$ instances fails to prove AND_2: Consider instances M_1 and M_2 with target t_1 and t_2, respectively, with $t_1 < t_2$. Both instances are no-instances in that M_1 overachieves, i.e., $\max(M_1) > t_1$, and M_2 underachieves, i.e., $\max(M_2) < t_2$. However, the maximum of each instance equals the target of the other instance, that is, $\max(M_1) = t_2$ and $\max(M_2) = t_1$. If we trivially merged both instances, we would have a yes-instance with a greatest Nash product of $t_1 \cdot t_2$, the target of the merger. Therefore, we preprocess both input instances with a polynomial-time algorithm.

4.2 Utilities Represented by Straight-Line Programs

When utilities are represented by straight-line programs, we prove NP-completeness for egalitarian social welfare optimization and social welfare optimization by the Nash product. This helps to complete the picture for the complexity of social welfare optimization problems with straight-line program representation of utility functions, which Dunne et al. [9] initiated by their NP-completeness result for utilitarian social welfare optimization.

Theorem 11 $\mathbb{Q}$-ESWO$_{\text{SLP}}$ *and* $\mathbb{Q}^{+}$-NPSWO$_{\text{SLP}}$ *are* NP*-complete.*

Proof. Membership in NP is easy to see. To show NP-hardness, we reduce from the NP-complete problem MAX3SAT, which is formally defined as follows:

MAX3SAT	
Given:	A boolean formula φ in 3-CNF (i.e., in conjunctive normal form with three literals per clause) and $k \geq 2$.
Question:	Is there an assignment to the variables of φ such that at least k clauses are satisfied?

Let $\varphi = \bigwedge_{i=1}^{m}(z_i^1 \vee z_i^2 \vee z_i^3)$ be a given boolean formula in 3-CNF, where z_i^j, $1 \leq i \leq m$ and $j \in \{1,2,3\}$, is a literal of some variable $v \in V = \{v_1,\ldots,v_n\}$. Define $A = \{a_1,a_2\}$ and $R = \{r_1,\ldots,r_n,r_{n+1},\ldots,r_{2n}\}$. We say a bundle $S \subseteq R$ or its corresponding vector $\alpha_S = (x_1,\ldots,x_n,x_{n+1},\ldots,x_{2n})$ is *valid* if

$$\bigwedge_{i=1}^{n} XOR(x_i,x_{n+i}) = \bigwedge_{i=1}^{n}(\neg x_i \wedge x_{n+i}) \vee (x_i \wedge \neg x_{n+i}) = 1,$$

i.e., *XOR* denotes the boolean exclusive-or operation. Define a_1's utility function as

$$u_1(S) = \begin{cases} \text{number of satisfied clauses in } \varphi \text{ by } S & \text{if } S \text{ is valid} \\ 0 & \text{otherwise} \end{cases}$$

and a_2's utility function as

$$u_2 \equiv \begin{cases} m & \text{if we reduce to the egalitarian social welfare} \\ 1 & \text{if we reduce to social welfare by the Nash product.} \end{cases}$$

Write

$$u_1(\alpha_S) = \left(\bigwedge_{i=1}^{n} XOR(x_i,x_{n+i})\right) \cdot \sum_{i=1}^{m}(z_i^1 \vee z_i^2 \vee z_i^3),$$

where we replace[4] z_i^j by the corresponding value of x_k, $k \in \{1,\ldots,n\}$, if z_i^j is a positive literal of x_k; otherwise (that is, if z_i^j is a negated variable) we replace it by the value of x_{n+k}, $k \in \{1,\ldots,n\}$. By Proposition 1, we know there is an SLP of polynomial size that represents u_1.

Now consider a 3-CNF formula φ whose maximum number of satisfied clauses is k for some assignment $A : X \to \{0,1\}$. Assignment A induces an assignment vector $\alpha_S = (A(v_1),\ldots,A(v_n),1 - A(v_1),\ldots,1 - A(v_n))$. By definition, α_S is valid and a_1's utility is exactly k. The remaining resources go to a_2. Because a_2's utility can be ignored, the social welfare is a_1's utility, that is, the maximum number of satisfied clauses in φ.

For the other direction, note that we reduced from a legal 3-CNF formula. So there is an assignment that satisfies at least one clause. Hence, a_1 realizes a utility of at least one. Now let $k \geq 1$ be the maximum social welfare of this instance. By definition, $u_1(S) = k$ for some valid $\alpha_S = (x_1,\ldots,x_n,x_{n+1},\ldots,x_{2n})$. Truncating α_S by dropping the last n coordinates yields an assignment that satisfies k clauses. ❑

5 Approximability of Social Welfare Optimization

In the previous section, we have shown that the decision versions of certain social welfare optimization problems are intractable: either NP-complete for the standard problem that asks whether a given threshold of social welfare can be reached or exceeded in a given MARA setting, or DP-complete for the exact variant. It is natural to ask whether the optimization problems corresponding to these decision problems are intractable as well, or whether they allow efficient approximation schemes.

Known approximability and inapproximability results in multiagent resource allocation have been surveyed recently in [19]. Here we prove some novel results not included there. The first one is an inapproximability result about social welfare optimization by the Nash product for 1-additive utilities. We prove this result by a reduction from the well-known NP-complete problem EXACT COVER BY THREE SETS, which is defined as follows:

[4]Because we have a boolean circuit, we actually insert an edge (x_k, o_p^q), where o_p^q, $p \in \{1,\ldots,m\}$, $q \in \{1,2\}$, denotes the $\vee$-gate that is responsible for z_i^j in clause p.

Exact Cover By Three Sets (X3C)	
Given:	A finite set B with $\|B\| = 3n$ and a collection $C = \{S_1, \dots, S_m\}$ of 3-element subsets of B.
Question:	Does there exist a subcollection $C' \subseteq C$ such that every element of B occurs in exactly one of the sets in C'?

Theorem 12 *Assuming* $\mathrm{P} \neq \mathrm{NP}$, $\textsc{Max}\text{-NPSW}_{\text{1-add}}$ *cannot be approximated within a factor of* $2/3 + \varepsilon$ *for any* $\varepsilon > 0$.

Proof. Let (B,C) with $\|B\| = 3n$ and $C = \{S_1, \dots, S_m\}$ be an instance of X3C. Without loss of generality, assume that $m \geq n$. Construct an instance $M = (A,R,U)$ of $\mathbb{Q}^+$-$\textsc{Max}\text{-NPSW}_{\text{1-add}}$ as follows. Let A be a set of m agents, where agent a_i corresponds to S_i, and let $R = B \cup D$ be a set of $2n+m$ resources. That is, there are $3n$ "real" resources that correspond to the $3n$ elements of B, and there are $m-n$ "dummy" resources in D. Define the agents' utilities as follows. For each $a_i \in A$ and each $r_j \in R$, let

$$u_i(r_j) = \begin{cases} 1/3 & \text{if } r_j \in S_i \\ 1 & \text{if } r_j \in D \\ 0 & \text{otherwise.} \end{cases}$$

Also, define $u_i(\emptyset) = 0$ for all i, $1 \leq i \leq m$.

Suppose that (B,C) is a yes-instance of X3C. Then there exists a set $I \subseteq \{1, \dots, m\}$, $\|I\| = n$, such that $S_i \cap S_j = \emptyset$ for all $i, j \in I$, $i \neq j$, and $\bigcup_{i \in I} S_i = B$. Hence, we assign the bundle S_i to agent a_i for each $i \in I$, and the dummy resources to the $m-n$ remaining agents. This allocation maximizes the Nash product social welfare, which now is at least 1. Furthermore, the sum of all agents' utilities is at most m. Hence, the product of the agents' individual utilities is maximal if and only if all agents have the same utility, which exactly equals 1.

Conversely, if (B,C) is a no-instance of X3C, we show that the maximum Nash product social welfare is at most $2/3$. Obviously, the sum of all agents' utilities is at most $m - 1/3$ in this case. The Nash product social welfare reaches the maximal value iff the utilities of the agents are as balanced as possible. The best allocation that satisfies this property is the following. Dummy resources are distributed to $m-n$ agents, $n-1$ agents get the $n-1$ disjoint bundles from $(S_1, \dots, S_m)$, and the last agent is assigned the remaining bundle which has utility of at most $2/3$. This implies that $\max_N(M) \leq 2/3$. Therefore, an approximation algorithm with a factor better than $2/3$ will distinguish the "yes" and "no" instances of X3C. ❑

Theorem 12 shows that $\textsc{Max}\text{-NPSW}_{\text{1-add}}$ cannot have a PTAS unless $\mathrm{P} = \mathrm{NP}$. This result also holds for $\textsc{Max}\text{-ESW}_{\text{1-add}}$ due to Bezáková and Dani [2]. However, we show that there is an FPTAS for this problem whenever the number of agents is fixed, using a technique that was also used to give an FPTAS for a variety of scheduling problems (see [26] and [13]). We assume that for any agent a_i, the utility function u_i is nonnegative and $u_i(\emptyset) = 0$. The proof is omitted as well.

Theorem 13 *Both* $\textsc{Max}\text{-NPSW}_{\text{1-add}}$ *and* $\textsc{Max}\text{-ESW}_{\text{1-add}}$ *admit an FPTAS for any fixed number of agents.*

Coming back to the straight-line program representation of utility functions, notice that the reduction in the proof of Theorem 11 is an L-reduction with parameters $\alpha = \beta = 1$. There is a one-to-one correspondence between assignments of variables and assignments of resources to the first agent, where the maximum number of satisfied clauses equals the social welfare after the reduction. By setting the utility function of the second agent to the constant zero-function, we have a reduction with the same properties for the utilitarian case. Using the inapproximability result for Max3SAT by Håstad [12], we conclude:

Corollary 14 $\mathbb{Q}$-MAX-USW$_{\text{SLP}}$, $\mathbb{Q}$-MAX-ESW$_{\text{SLP}}$, $\mathbb{Q}^{+}$-MAX-NPSW$_{\text{SLP}}$ *are* NP*-hard to approximate within a factor of* $7/8 + \varepsilon$ *for every* $\varepsilon > 0$.

6 Conclusions

We have classified the decision versions of social welfare optimization problems for egalitarian and Nash product social welfare (for utilities represented by straight-line programs) and the exact variant for Nash product social welfare (for utilities in the bundle form and in the k-additive form) in terms of computational complexity. In addition, we have shown new approximability and inapproximability results for utilitarian, egalitarian, and Nash product social welfare. As interesting open problems for future work, we mention the study of complexity and approximability of social welfare optimization problems for different representation forms, improving approximation algorithms, and identifying tractable cases of restricted problem variants.

Acknowledgments: We gratefully acknowledge interesting discussions with Ulle Endriss, and we thank the COMSOC-2012, AAMAS-2012, ISAIM-2012, and STAIRS-2012 reviewers for their helpful comments.

References

[1] S. Arora and C. Lund. Hardness of approximations. In D. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 10, pages 399–446. PWS Publishing Company, 1996.

[2] I. Bezáková and V. Dani. Allocating indivisible goods. *SIGecom Exchanges*, 5(3):11–18, 2005.

[3] S. Bouveret. *Fair Allocation of Indivisible Items: Modeling, Computational Complexity and Algorithmics*. PhD thesis, Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France, November 2007.

[4] R. Chang and J. Kadin. On computing boolean connectives of characteristic functions. *Theory of Computing Systems*, 28(3):173–198, 1995.

[5] Y. Chevaleyre, P. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica*, 30:3–31, 2006.

[6] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation with k-additive utility functions. In *Proceedings of the DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, volume 3 of *Annales du LAMSADE*, pages 83–100, 2004.

[7] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation in k-additive domains: Preference representation and complexity. *Annals of Operations Research*, 163:49–62, 2008.

[8] V. Conitzer, T. Sandholm, and P. Santi. Combinatorial auctions with k-wise dependent valuations. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 248–254. AAAI Press, 2005.

[9] P. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 164(1–2):23–46, 2005.

[10] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[11] M. Grabisch. k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets and Systems*, 92(2):167–189, 1997.

[12] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[13] E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *Journal of the ACM*, 23(2):317–327, 1976.

[14] R. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 125–131. ACM Press, 2004.

[15] H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2004.

[16] N. Nguyen, T. Nguyen, M. Roos, and J. Rothe. Complexity and approximability of egalitarian and Nash product social welfare optimization in multiagent resource allocation. In *Proceedings of the 6th European Starting AI Researcher Symposium*. IOS Press, August 2012. To appear.

[17] N. Nguyen, T. Nguyen, M. Roos, and J. Rothe. Complexity and approximability of social welfare optimization in multiagent resource allocation (extended abstract). In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1287–1288. IFAAMAS, June 2012.

[18] N. Nguyen, M. Roos, and J. Rothe. Exact optimization of social welfare by the Nash product is DP-complete. In *Website Proceedings of the 12th International Symposium on Artificial Intelligence and Mathematics*, January 2012.

[19] T. Nguyen, M. Roos, and J. Rothe. A survey of approximability and inapproximability results for social welfare optimization in multiagent resource allocation. In *Website Proceedings of the Special Session on Computational Social Choice at the 12th International Symposium on Artificial Intelligence and Mathematics*, January 2012.

[20] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, second edition, 1995.

[21] C. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244–259, 1984.

[22] N. Pippenger and M. Fischer. Relations among complexity measures. *Journal of the ACM*, 26(2):361–381, 1979.

[23] S. Ramezani and U. Endriss. Nash social welfare in multiagent resource allocation. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, pages 117–131. Springer-Verlag *Lecture Notes in Business Information Processing #79*, 2010.

[24] M. Roos and J. Rothe. Complexity of social welfare optimization in multiagent resource allocation. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 641–648. IFAAMAS, May 2010.

[25] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2005.

[26] S. Sahni. Algorithms for scheduling independent tasks. *Journal of the ACM*, 23(1):116–127, 1976.

[27] C. Schnorr. The network complexity and the Turing machine complexity of finite functions. *Acta Informatica*, 7(1):95–107, 1976.

[28] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, second edition, 2003.

Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: nhan-tam.nguyen@uni-duesseldorf.de
{thanh,roos,rothe}@cs.uni-duesseldorf.de

On the Complexity of Voting Manipulation under Randomized Tie-Breaking

Svetlana Obraztsova   Yair Zick   Edith Elkind

Abstract

Computational complexity of voting manipulation is one of the most actively studied topics in the area of computational social choice, starting with the groundbreaking work of Bartholdi et al. [2]. Most of the existing work in this area, including that of Bartholdi et al., implicitly assumes that whenever several candidates receive the top score with respect to the given voting rule, the resulting tie is broken according to a lexicographic ordering over the candidates. In this paper, we explore an equally appealing method of tie-breaking, namely, selecting the winner uniformly at random among all tied candidates. We show that under this method of breaking ties, all scoring rules, the Bucklin rule and Plurality with Runoff remain easy to manipulate; however, finding a manipulative vote becomes NP-hard for Copeland and Maximin. We extend some of our easiness results to elections with multiple winners. We show that if the number of winners is small, then manipulation is in P for all scoring rules, and it is in P for k-Approval for any number of winners.

1 Introduction

Whenever a group of agents have to make a joint decision, the agents' opinions need to be aggregated in order to identify a suitable course of action. This applies both to human societies and to groups of autonomous agents; the entities that the agents need to select from vary from political leaders to song contest winners and joint plans. The standard way to aggregate preferences is by asking the agents to vote over the available candidates: each agent ranks the candidates, and a *voting rule*, i.e., a mapping from collective rankings to candidates, is used to select the winner.

In most preference aggregation settings, each agent wants his most favorite alternative to win, irrespective of other agents' preferences. Thus, he may try to *manipulate* the voting rule, i.e., to misrepresent his preferences in order to obtain an outcome that he ranks higher than the outcome of the truthful voting. Indeed, the famous Gibbard–Satterthwaite theorem [9, 15] shows that whenever the agents have to choose from 3 or more alternatives, every reasonable voting rule is manipulable, i.e., for some collection of voter's preferences some voter can benefit from lying about his ranking. This is bad news, as the manipulator may exercise undue influence over the election outcome, and a lot of research effort has been invested in identifying voting rules that are more resistant to manipulation than others, as measured by the fraction of manipulable profiles or the algorithmic complexity of manipulation (see [7] for an overview).

Many common voting rules operate by assigning scores to candidates, so that the winner is the candidate with the highest score. Now, in elections with a large number of voters and a small number of candidates there is usually only one candidate that obtains the top score. However, this does not necessarily hold when the alternative space is large, as may be the case when, e.g., agents in a multiagent system use voting to decide on a joint plan of action [6]. If, nevertheless, a single outcome needs to be selected, such ties have to be broken. In the context of manipulation this means that the manipulator should take the tie-breaking rule into account when choosing his actions. Much of the existing literature on voting manipulation circumvents the issue by assuming that the manipulator's goal is to make some distinguished candidate p one of the election winners, or, alternatively, the unique winner. The former assumption can be interpreted as a tie-breaking rule that is favorable to the manipulator, i.e., given a tie that involves p, always selects p as the winner; similarly, the latter assumption corresponds to a tie-breaking rule that is adversarial to the manipulator. In fact, most of the existing algorithms for finding a manipulative vote work for any tie-breaking rule that selects the winner according to a given ordering on the candidates; the two cases considered above correspond to this order being, respectively, the manipulator's preference order or its inverse.

However, till recently, an equally appealing approach to tie-breaking, namely, selecting the winner among all tied candidates uniformly at random, has been rarely studied. Two exceptions to this pattern that we are aware of are [11] and [5]; however, [11] does not deal with manipulation at all, while [5] considers a very different model of manipulation. Perhaps one of the reasons for this is that under randomized tie-breaking the outcome of the election is a random variable, so it is not immediately clear how to compare two outcomes: is having your second-best alternative as the only winner preferable to the lottery in which your top and bottom alternatives have equal chances of winning?

We deal with this issue by augmenting the manipulator's preference model: we assume that the manipulator assigns a numeric utility to all candidates, and his goal is to vote so as to maximize his expected utility, where the expectation is computed over the random choices of the tie-breaking procedure; this approach is standard in the social choice literature (see, e.g., [10]) and has also been used in [5]. We show that in this model all scoring rules are easy to manipulate; this is also the case for Bucklin (both for its classic and simplified versions). On the other hand, we prove that manipulation under randomized tie-breaking is hard for Maximin and Copeland. We complement these hardness results by identifying a natural assumption on the manipulator's utility function that makes Maximin easy to manipulate. We also analyze the complexity of manipulation for three voting rules that compute the winners using a multi-step procedure, namely, Plurality with Runoff, STV, and Ranked Pairs. Thus, we provide an essentially complete picture of the complexity of manipulating common voting rules under randomized tie-breaking (see Table 1 in the end of the paper). Finally, we explore the complexity of manipulation when voters need to choose several winners. We show that for the k-Approval voting rule, multi-winner manipulation is in P; moreover, if the number of winners to be selected is small (i.e., bounded by a constant), then manipulating an election under any scoring rule is also in P.

Some of the results that appear in this paper were previously published in [14] and [13]; however, the material in Section 5 is new.

2 Preliminaries

An *election* is given by a set of candidates $C = \{c_1, \dots, c_m\}$ and a vector $\mathcal{R} = (R_1, \dots, R_n)$, where each R_i, $i = 1, \dots, n$, is a linear order over C; R_i is called the *preference order* (or, *vote*) of voter i. We denote the space of all linear orderings over C by $\mathcal{L}(C)$. The vector $\mathcal{R} = (R_1, \dots, R_n)$ is called a *preference profile*. For readability, we will sometimes denote R_i by $\succ_i$. When $a \succ_i b$ for some $a, b \in C$, we say that voter i prefers a to b. We denote by $r(c_j, R_i)$ the *rank* of candidate c_j in the preference order R_i: $r(c_j, R_i) = |\{c \in C \mid c \succ_i c_j\}| + 1$.

A *voting rule* $\mathcal{F}$ is a mapping that, given a preference profile $\mathcal{R}$ over C, outputs a candidate $c \in C$; we write $c = \mathcal{F}(\mathcal{R})$. Many classic voting rules, such as the ones defined below, are, in fact, *voting correspondences*, i.e., they map a preference profile $\mathcal{R}$ to a non-empty subset S of C. Voting correspondences can be transformed into voting rules using *tie-breaking rules*.

A tie-breaking rule for an election $(C, \mathcal{R})$ is a mapping $T = T(\mathcal{R}, S)$ that for any $S \subseteq C$, $S \neq \emptyset$, outputs a candidate $c \in S$. We say that a tie-breaking rule T is *lexicographic* with respect to a preference ordering $\succ$ over C if for any preference profile $\mathcal{R}$ over C and any $S \subseteq C$ it selects the most preferred candidate from S with respect to $\succ$, i.e., we have $T(S) = c$ if and only if $c \succ a$ for all $a \in S \setminus \{c\}$.

A *composition* of a voting correspondence $\mathcal{F}$ and a tie-breaking rule T is a voting rule $T \circ \mathcal{F}$ that, given a preference profile $\mathcal{R}$ over C, outputs $T(\mathcal{R}, \mathcal{F}(\mathcal{R}))$. Clearly, $T \circ \mathcal{F}$ is a voting rule and $T \circ \mathcal{F}(\mathcal{R}) \in \mathcal{F}(\mathcal{R})$.

Voting Rules We now describe the voting rules (correspondences) considered in this paper. For all rules that assign scores to the candidates the winners are the candidates with the highest scores.

Scoring rules: Any vector $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathbb{R}^m$ with $\alpha_1 \geq \dots \geq \alpha_m$ defines a *scoring rule* $\mathcal{F}_\alpha$. Under this rule, each voter grants α_i points to the candidate it ranks in the i-th position; the score of a candidate is the sum of the scores it receives from all voters. The vector α is called a *scoring vector*. A well-known example of a family of scoring rules is *Borda*, given by $\alpha^m = (m-1, \dots, 1, 0)$; another example is k-Approval, where a candidate gets one point for each voter that ranks him in the top k positions. 1-Approval is also known as *Plurality*.

Bucklin: Let k^* be the smallest k such that the k-approval score of some $c \in C$ is at least $\lfloor n/2 \rfloor + 1$; we say that k^* is the *Bucklin winning round*. Given a candidate $c \in C$, his *Bucklin score* is his k^*-approval score. Under the *simplified Bucklin* rule, candidates whose Bucklin score is at least $\lfloor n/2 \rfloor + 1$ are the winners, while under *Bucklin*, winners are those with the highest Bucklin score.

Copeland: We say that a candidate a wins a *pairwise election* against b if more than half of the voters prefer a to b; if exactly half of the voters prefer a to b, then a is said to *tie* his pairwise election against b. Given a rational value $\alpha \in [0, 1]$, under the Copeland$^\alpha$ rule each candidate gets 1 point for each pairwise election he wins and α points for each pairwise election he ties.

Maximin: For every pair of candidates $c, d \in C$, we set $s(c, d) = |\{i \mid c \succ_i d\}|$. The Maximin score of a candidate $c \in C$ is given by $\min_{d \in C \setminus \{c\}} s(c, d)$; that is, c's Maximin score is the number of votes he gets in his worst pairwise election.

Plurality with Runoff and STV: Under the STV rule, the election proceeds in rounds; in each round, the candidate with the lowest Plurality score is eliminated, and candidates' scores are recomputed. The winner is the candidate that survives till the last round. Plurality with Runoff can be thought of as a compressed version of STV: we first select two candidates with the highest Plurality scores, and then output the winner of the pairwise election between them. Note that these definitions are somewhat ambiguous, as several candidates may have the lowest/highest Plurality score; we will comment on this issue in Section 4.

3 The Model

Given a preference profile $\mathcal{R}$ over a candidate set C and a preference order L over C, let $(\mathcal{R}_{-i}, L)$ be the preference profile obtained from $\mathcal{R}$ by replacing R_i with L. We say that a voter $i \in \{1, \dots, n\}$ can successfully *manipulate* an election $(C, \mathcal{R})$ with respect to a voting rule $\mathcal{F}$ if $\mathcal{F}(\mathcal{R}_{-i}, L) \succ_i \mathcal{F}(\mathcal{R})$. We will now explain how to extend this definition to voting correspondences under the assumption that ties are broken uniformly at random.

Given a voting correspondence $\mathcal{F}$ and an election $(C, \mathcal{R})$, suppose that $\mathcal{F}(C, \mathcal{R}) = S$, where $|S| > 1$. Suppose that we select the winner uniformly at random, i.e., every candidate in S has the same chance of being selected. In this case, knowing the manipulator's preference ordering is not sufficient to determine his optimal strategy. For example, suppose that the manipulator prefers a to b to c, and by voting strategically he can change the output of $\mathcal{F}$ from b to $\{a, c\}$. It is not immediately clear if this manipulation is beneficial. Indeed, if the manipulator strongly prefers a, but is essentially indifferent between b and c, then the answer is probably positive, but if he strongly dislikes c and slightly prefers a to b, the answer is likely to be negative (of course, this also depends on the manipulator's risk attitude).

Thus, to model this situation appropriately, we need to know the utilities that the manipulator assigns to all candidates. Under the assumption of risk neutrality, the manipulator's utility for a set of candidates is equal to his expected utility when a candidate is drawn from this set uniformly at random, or, equivalently, to his *average* utility for a candidate in this set. Since we are interested in computational issues, we assume that all utilities are positive integers given in binary.

Formally, given a set of candidates C, we assume that the manipulator is endowed with a utility function $u : C \to \mathbb{N}$. This function can be extended to sets of candidates by setting $u(S) = \frac{1}{|S|}\sum_{c \in S} u(c)$ for any $S \subseteq C$.

3.1 Single-Winner Elections

We now define the manipulation problem in the single-winner case. As all voting rules considered in this paper are anonymous, we can fix any voter as the manipulator. In what follows, it will be convenient to assume that the manipulator is voter n.

Definition 3.1. *An instance of the $\mathcal{F}$-RANDMANIPULATION problem is a tuple (E, u, q), where $E = (C, \mathcal{R})$ is an election, $u : C \to \mathbb{N}$ is the manipulator's utility function such that $u(c) \geq u(c')$ if and only if $c \succ_n c'$, and q is a non-negative rational number. It is a "yes"-instance if there exists a vote L such that $u(\mathcal{F}(\mathcal{R}_{-n}, L)) \geq q$ and a "no"-instance otherwise.*

The optimization version of $\mathcal{F}$-RANDMANIPULATION is defined similarly. We remark that $\mathcal{F}$-RANDMANIPULATION is in NP for any polynomial-time computable voting correspondence $\mathcal{F}$: it suffices to guess the manipulative vote L, determine the set $S = \mathcal{F}(\mathcal{R}_{-n}, L)$, and compute the average utility of the candidates in S.

3.2 Multi-Winner Elections

There are settings where voters elect more than one candidate. In that case, ℓ members of C will be named the winners, or the *elected committee*. There are many voting rules that are designed specifically for this setting and aim to select the candidates that best represent the voters (see e.g. Chamberlin and Courant [3]). However, in this paper, we will focus on using scoring rules for the purpose of manipulation; this approach is reasonable when voting is used to select the finalists of a contest, or to allocate grants or fellowships (see, for example the work by Meir et al. [12]). Given a scoring rule, if we want to elect a committee of size ℓ, it is natural to choose the ℓ candidates with the highest score. However, we may still need to break ties. Suppose, for instance, that $\ell = 3$ and we have two candidates whose score is 10 and two candidates whose score is 9. Clearly, the candidates whose score is 10 should be elected no matter what, but we need to choose one of the candidates whose score is 9, e.g., by tossing a fair coin. We will now explain how to formalize this approach.

Fix a scoring rule $\mathcal{F}_\alpha$ with $\alpha = (\alpha_1, \ldots, \alpha_m)$ and an election $E = (C, \mathcal{R})$ with $|C| = m$. Given a candidate $c \in C$, let s_c denote c's score in E under $\mathcal{F}_\alpha$. We say that candidates c and c' are on the same *level* if $s_c = s_{c'}$. There are $p \leq m$ levels, denoted $H_1, \ldots, H_p$; we set $s(H_j)$ to be the score of the candidates in H_j, and assume that $s(H_1) > \ldots > s(H_p)$. Let $W_j = \cup_{q=1}^{j} H_q$. If $|W_j| \leq \ell$, then the tie-breaking rule does not apply to W_j. Formally, let $j_0 = \max\{j \mid |W_j| \leq \ell\}$ and set $\mathcal{W} = W_{j_0}$. The set $\mathcal{W}$ is called the *confirmed set*: these are the candidates who will definitely be in the elected committee. The set $\mathcal{P} = H_{j+1}$ is called the *pending set*: these are the candidates to which we must apply the tie-breaking rule. Note that $|H_1| > \ell$ implies $\mathcal{W} = \emptyset$ and $\mathcal{P} = H_1$, and $|\mathcal{W}| = \ell$ implies $\mathcal{P} = \emptyset$. For single-winner elections ($\ell = 1$) we obtain $\mathcal{P} = \emptyset$ if $|H_1| = 1$, and $\mathcal{P} = H_1$ otherwise. The randomized tie-breaking rule operates by choosing $\ell' = \ell - |\mathcal{W}|$ candidates from the set $\mathcal{P}$ uniformly at random.

We assume that the manipulator's utility is additive, i.e., if a committee $S \subseteq C$ is elected, his utility is given by $\sum_{c \in S} u(c) = |S|u(S)$. Let $T_s(\mathcal{P})$ denote the random variable that takes values in the space of all s-subsets of $\mathcal{P}$, with each subset being equally likely. Given a variable ξ, let $\mathbb{E}[\xi]$ denote its expectation. Then the manipulator's utility in E (under truthful voting) is

$$\sum_{c \in \mathcal{W}} u(c) + \mathbb{E}[\sum_{c \in T_{\ell'}(\mathcal{P})} u(c)],$$

where $\ell' = \ell - |\mathcal{W}|$.

We are now ready to define the computational problem that is associated with manipulating a multi-winner election under randomized tie-breaking with respect to a scoring rule $\mathcal{F}_\alpha$, $\alpha = (\alpha_1, \dots \alpha_m)$. We will refer to this problem as $\mathcal{F}_\alpha$-RANDMULTIMANIPULATION. An instance of this problem is given by an election $E = (C, \mathcal{R})$ with $|C| = m$, a committee size ℓ, the manipulator's utility function $u : C \to \mathbb{N}$, which satisfies $u(c) \geq u(c')$ if and only if $c \succ_n c'$, and a non-negative rational number q. It is a "yes"-instance if there exists a vote L such that the manipulator's utility in $(C, (\mathcal{R}_{-n}, L))$ is at least q and a "no"-instance otherwise.

4 Single-Winner Elections

We begin by analyzing the family of scoring rules. We observe that for any scoring rule, manipulation is easy under randomized tie-breaking.

Theorem 4.1. *For any scoring vector* $\alpha = (\alpha_1, \dots, \alpha_m)$ $\mathcal{F}_\alpha$-RANDMANIPULATION *is in* P.

Theorem 4.1 can be obtained as a corollary of Theorem 5.2 in Section 5 (see [14] for a direct proof). It implies that for scoring rules, assuming that ties are broken uniformly at random does not increase the complexity of manipulation compared to lexicographic tie-breaking.

Similarly, both the classic and the simplified versions of the Bucklin rule can be manipulated in polynomial time; the proof is omitted due to space constraints.

Theorem 4.2. Bucklin-RANDMANIPULATION *and* simplified Bucklin-RANDMANIPULATION *are in* P.

In contrast, if we break ties uniformly at random, manipulation under Maximin becomes NP-hard. In fact, our hardness result holds even for a fairly simple utility function: Let w be the Maximin winner prior to the manipulators vote; if we set $u(w) = 0$, $u(c) = 1$ for $c \in C \setminus \{w\}$, then Maximin-RANDMANIPULATION becomes NP-complete. The proof is omitted due to space constraints.

Theorem 4.3. Maximin-RANDMANIPULATION *is* NP*-complete.*

While Maximin-RANDMANIPULATION is NP-complete in general, there is an efficient algorithm for this problem assuming that the manipulator's utility function has special structure. Specifically, recall that in the model of [2] the manipulator's goal is to make a specific candidate p a winner. This suggests that the manipulator's utility can be modeled by setting $u(p) = 1$, $u(c) = 0$ for all $c \in C \setminus \{p\}$. We will now show that for such utilities, Maximin-RANDMANIPULATION is in P.

Theorem 4.4. *If the manipulator's utility function is given by* $u(p) = 1$, $u(c) = 0$ *for* $c \in C \setminus \{p\}$, *Maximin-*RANDMANIPULATION *is in* P.

Proof. Consider an election $E = (C, \mathcal{R})$ with the candidate set $C = \{c_1, \dots, c_m\}$ and recall that n is the manipulating voter. In this proof, we denote by $s(c_i)$ the Maximin score of a candidate $c_i \in C$ in the election $E' = (C, \mathcal{R}')$, where $\mathcal{R}' = \mathcal{R}_{-n}$. Let $s = \max_{c_i \in C} s(c_i)$.

For any $c_i \in C$, the manipulator's vote increases the score of c_i either by 0 or by 1. Thus, if $s(p) < s - 1$, the utility of the manipulator will be 0 irrespective of how he votes.

Now, suppose that $s(p) = s - 1$. The manipulator can increase the score of p by 1 by ranking p first. Thus, his goal is to ensure that after he votes (a) no other candidate gets $s + 1$ point and (b) the number of candidates in $C \setminus \{p\}$ with s points is as small as possible. Similarly, if $s(p) = s$, the manipulator can ensure that p gets $s + 1$ points by ranking him first, so his goal is to rank the remaining candidates so that in $C \setminus \{p\}$ the number of candidates with $s + 1$ points is as small as possible. We will now describe an algorithm that works for both of these cases.

We construct a directed graph G with the vertex set C that captures the relationship among the candidates. Namely, we have an edge from c_i to c_j if there are $s(c_j)$ votes in $\mathcal{R}'$ that rank c_j above

c_i. Observe that, by construction, each vertex in G has at least one incoming edge. We say that c_i is a *parent* of c_j in G whenever there is an edge from c_i to c_j. We remark that if the manipulator ranks one of the parents of c_j above c_j in his vote, then c_j's score does not increase. We say that a vertex c_i of G is *purple* if $s(c_i) = s(p) + 1$, *red* if $s(c_i) = s(p)$ and $c_i \neq p$, and *green* otherwise; note that by construction p is green. Observe also that if $s(p) = s$, there are no purple vertices in the graph. We will say that a candidate c_j is *dominated* in an ordering L (with respect to G) if at least one of c_j's parents in G appears before c_j in L. Thus, our goal is to ensure that the set of dominated candidates includes all purple candidates and as many red candidates as possible.

Our algorithm is based on a recursive procedure $\mathcal{A}$, which takes as its input a graph H with a vertex set $U \subseteq C$ together with a coloring of U into green, red and purple; intuitively, U is the set of currently unranked candidates. It returns "no" if the candidates in U cannot be ranked so that all purple candidates in U are dominated by other candidates in U with respect to H. Otherwise, it returns an ordered list L of the candidates in U in which all purple candidates are dominated, and a set S consisting of all red candidates in U that remain undominated in L with respect to H.

To initialize the algorithm, we call $\mathcal{A}(G)$. The procedure $\mathcal{A}(H)$ is described below (Algorithm 1). We claim that $\mathcal{A}(G)$ outputs "no" if and only if no matter how the manipulator votes, some candidate in $C \setminus \{p\}$ gets $s(p) + 2$ points. Moreover, if $\mathcal{A}(G) = (L, S)$ and the set S contains r red candidates, then for any vote of the manipulator that ensures that all candidates in $C \setminus \{p\}$ have at most $s(p) + 1$ points there are at least r red candidates with $s(p) + 1$ points. We will split the proof of this claim into several lemmas, whose proofs are omitted.

Lemma 4.5. *At any point in the execution of the algorithm, if $\mathcal{A}(H) = (L, S)$, then each candidate in $U \setminus S$ is dominated in H.*

We are now ready to prove that our algorithm correctly determines whether the manipulator can ensure that no candidate gets more than $s(p) + 1$ points.

Lemma 4.6. *The algorithm outputs "no" if and only if for any vote L there is a purple candidate that is undominated.*

It remains to show that the set S output by the algorithm contains as few candidates as possible.

Lemma 4.7. *At any point in the execution of Algorithm 1, if $\mathcal{A}(H) = (L, S)$, then in any ordering of the candidates in U in which each purple vertex in U is dominated, at least $|S|$ red vertices in U are undominated.*

Combining Lemma 4.6 and Lemma 4.7, we conclude that Algorithm 1 outputs (L, S), then L is the optimal vote for the manipulator and if Algorithm 1 outputs "no", then the manipulator's utility is 0 no matter how he votes. Also, it is not hard to see that Algorithm 1 runs in polynomial time. □

We remark that Theorem 4.4 has recently been extended to utility functions that assign utility of 1 to a constant number of candidates and utility of 0 to all other candidates (see [17]).

For the Copeland rule, RANDMANIPULATION is also NP-hard. To show this, we give a reduction from the INDEPENDENT SET problem [8] (proof omitted due to space constraints).

Theorem 4.8. $\mathrm{Copeland}^{\alpha}$-RANDMANIPULATION *is* NP*-complete for any rational* $\alpha \in [0, 1]$.

Some common voting rules, such as, e.g., STV, do not assign scores to candidates; rather, they are defined via multi-step procedures. When one computes the winner under such rules, ties may have to be broken during each step of the procedure. A natural approach to winner determination under such rules is to use the *parallel universes tie-breaking* [4]: a candidate c is an election winner if the intermediate ties can be broken so that c is a winner after the final step. Thus, any such rule defines a voting correspondence in the usual way, and hence the corresponding RANDMANIPULATION problem is well-defined. In this paper we consider three rules in this class, namely, Plurality with Runoff, STV, and Ranked Pairs (we use the definition in [4]).

Algorithm 1: $\mathcal{A}(H)$

```
L ← ∅;
if H contains p then
    L ← [p]; H ← H \ {p};
while H contains a candidate c that is green or has a parent that has already been ranked in
the input graph H do
    L ← L ::[c]; H ← H \ {c};
// :: is the string append operation.
if H = ∅ then
    return (L, ∅);
if there is a purple candidate in H with no parents in H then
    return "no";
if there is a red candidate c in H with no parents in H then
    H′ ← H with c colored green;
    OUT ← A(H′);
    if OUT ="no" then
        return "no"
    (L′, S′) ← OUT;
    return (L::L′, S′ ∪ {c}).
Let T be some cycle in H;
// At this point in the algorithm, each vertex of H has a
   parent, thus there is a cycle in H
Collapse T;
// i.e., (a) replace T with a single vertex t, and (b) for
   each y ∉ T, add an edge (t, y) if H contained an edge (x, y) for
   some x ∈ T and add an edge (y, t) if H contained a vertex z
   with (y, z) ∈ H
if T contains at least one red vertex then
    color t red;
else
    color t purple;
H′ ← H after the contraction;
OUT ← A(H′);
if OUT = "no" then
    return "no";
(L′, S′) ← OUT;
if t ∈ S′ then
    // t must be red, so T contains a red vertex
    Let c be some red vertex in T;
    Let L̂ be an ordering of the vertices in T that starts with c and follows the edges of T;
    Let L″ be the list obtained from L′ by replacing t with L̂;
    return (L::L″, (S′ \ {t}) ∪ {c}).
else
    // t ∉ S, so by Lemma 4.5 t is dominated in H′
    Let a be a parent of t that precedes it in L′;
    Let c be some child of a that appears in T;
    Let L̂ be an ordering of the vertices in T that starts with c and follows the edges of T;
    Let L″ be the list obtained from L′ by replacing t wish L̂;
    return (L::L″, S′).
```

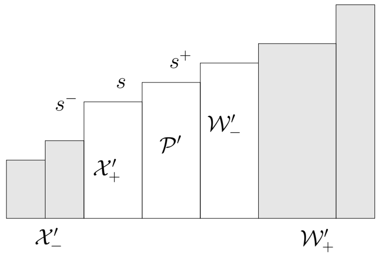


Figure 1: Proof of Theorem 5.1

Proposition 4.9. Plurality with Runoff-RANDMANIPULATION *is in* P.

We omit the proof of Proposition 4.9 due to space constraints; briefly, the main idea of the proof is that the best manipulative vote can be found by placing some candidate first and then ranking the remaining candidates according to their utility.

However, for STV and Ranked Pairs, RANDMANIPULATION is NP-hard. The proof of this fact hinges on an observation that allows us to inherit hardness results from the standard model of voting manipulation. Recall that $\mathcal{F}$-COWINNERMANIPULATION is the computational problem of deciding whether given an election $E = (C, \mathcal{R})$ the manipulator can make a specific candidate $p \in C$ one of the election winners under a voting correspondence $\mathcal{F}$.

Proposition 4.10. *For any voting correspondence $\mathcal{F}$, the problem $\mathcal{F}$-*COWINNERMANIPULATION *many-one reduces to $\mathcal{F}$-*RANDMANIPULATION.

Since for STV and Ranked Pairs COWINNERMANIPULATION is known to be NP-hard (see, respectively, [1] and [16]), we obtain the following corollary.

Corollary 4.11. STV-RANDMANIPULATION *and* Ranked Pairs-RANDMANIPULATION *are* NP-*hard.*

5 Multi-Winner Elections

We now discuss the complexity of manipulating multi-winner elections when ties are broken uniformly at random.

We begin by analyzing the k-Approval voting correspondence. It turns out that k-Approval-RANDMULTIMANIPULATION can be decided in polynomial time.

Theorem 5.1. k-Approval-RANDMULTIMANIPULATION *is in* P.

Proof. Consider the election $(C, \mathcal{R}_{-n})$, and let $\mathcal{P}'$ and $\mathcal{W}'$ be, respectively, the pending set and the confirmed set in this election. Set $\mathcal{X}' = C \setminus (\mathcal{P}' \cup \mathcal{W}')$. Let s^+ be the lowest k-Approval score among the candidates in $\mathcal{W}'$ (set $s^+ = +\infty$ if $\mathcal{W}' = \emptyset$), let s^- be the highest k-Approval score among the candidates in $\mathcal{X}'$ (set $s^- = -\infty$ if $\mathcal{X}' = \emptyset$), and let s be the k-Approval score of the candidates in $\mathcal{P}'$ (if $\mathcal{P}' = \emptyset$, s remains undefined). Let $\mathcal{W}'_- \subseteq \mathcal{W}'$ be the set of candidates whose k-Approval score is s^+, and let $\mathcal{X}'_+ \subseteq \mathcal{X}'$ be the set of candidates whose k-Approval score is s^-; also, set $\mathcal{W}'_+ = \mathcal{W}' \setminus \mathcal{W}'_-$ and $\mathcal{X}'_- = \mathcal{X}' \setminus \mathcal{X}'_+$. Note that $s^- < s^+$, and if $\mathcal{P}' \neq \emptyset$, we have $s^- < s < s^+$.

Let E be the election obtained after the manipulator votes, and suppose that in E the confirmed set is $\mathcal{W}$ and the pending set is $\mathcal{P}$; also, set $\mathcal{X} = C \setminus (\mathcal{W} \cup \mathcal{P})$. We will now argue that, no matter how the manipulator votes, we have $\mathcal{W}'_+ \subseteq \mathcal{W}$ and $\mathcal{X}'_- \subseteq \mathcal{X}$, i.e., points allocated to candidates in $\mathcal{W}'_+ \cup \mathcal{X}'_-$ do not affect the election outcome. Indeed, in E the score of every candidate in $\mathcal{W}'_+$ will be at least $s^+ + 1$, and there can be at most $|\mathcal{W}'| \leq \ell$ candidates with such score, so every candidate in $\mathcal{W}'_+$ will end up in $\mathcal{W}$. Further, in E the score of every candidate in $\mathcal{X}'_-$ will be at most s^-, and there are at least $|\mathcal{P}'| + |\mathcal{W}'| \geq \ell$ candidates whose score is at least $s^- + 1$, so the score of s^- will be insufficient for being placed in $\mathcal{P}$.

Now, suppose that the manipulator has decided to approve k_w candidates in $\mathcal{W}'_-$. Then, to maximize his utility, he has to approve k_w candidates in $\mathcal{W}'_-$ with the highest utility. A similar argument works for $\mathcal{P}'$ and $\mathcal{X}'_+$. As for the candidates in $\mathcal{W}'_+ \cup \mathcal{X}'_-$, it does not matter which ones he chooses to approve, since, as argued above, his vote will not change the status of these candidates. Thus, the outcome of the election is completely determined by a triple of non-negative integers (k_w, k_p, k_x), where k_w, k_p, and k_x are, respectively, the number of candidates in $\mathcal{W}'_-$, $\mathcal{P}'$, and $\mathcal{X}'_+$ that the manipulator approves. Hence, the manipulator can go over all triples of integers $(k_w, k_p, k_x) \in \{0, \dots, k\}^3$, and, for each triple, check if it corresponds to a valid vote and compute the expected utility that he obtains from approving k_w highest-utility candidates from $\mathcal{W}'_-$, k_p highest-utility candidates from $\mathcal{P}'$, and k_x highest-utility candidates from $\mathcal{X}'_+$, and distributing the remaining points (if any) among the rest of the candidates. The manipulator can then check if the expected utility from the best such triple is at least q. Clearly, (k_w, k_p, k_x) corresponds to a valid vote if and only if

- $0 \leq k_w \leq |\mathcal{W}'_-|$,
- $0 \leq k_p \leq |\mathcal{P}'|$,
- $0 \leq k_x \leq |\mathcal{X}'_+|$, and
- $0 \leq k - k_w - k_p - k_x \leq |\mathcal{X}'_-| + |\mathcal{W}'_+|$,

and the manipulator's expected utility from any such vote can be computed in time $O(k)$. Thus, the overall running time of our algorithm is $O(k^4)$. Since we can assume that $k \leq m$, this running time is polynomial in the input size. □

We now show that when the size of the committee, ℓ, is bounded by a constant, then $\mathcal{F}_\alpha$-RANDMULTIMANIPULATION is in P for any scoring rule F_α. This immediately implies the single winner case discussed in Section 4.

Theorem 5.2. *$\mathcal{F}_\alpha$-*RANDMULTIMANIPULATION *is in* P *when ℓ is bounded by a constant.*

Proof. Fix a scoring rule $\mathcal{F}_\alpha$ with a scoring vector $\alpha = \alpha_1 \geq \ldots \geq \alpha_m$ and an election $(C, \mathcal{R})$ with $|C| = m$, and let $\mathbf{s} = (s_1, \dots, s_m)$ be the vector of the candidates' scores in $(C, \mathcal{R}_{-n})$. For each $k \leq \ell$ and each subset $\mathcal{W}_k \subseteq C$ of size k, we check if the manipulator can vote so that the confirmed set is $\mathcal{W}_k$. If this is indeed the case, we find the best set of $\ell - k$ pending winners for this choice of $\mathcal{W}_k$; that is, we identify a set $\mathcal{P}_k$ with $|\mathcal{P}_k| > \ell - k$ such that after the manipulator's vote the confirmed set is $\mathcal{W}_k$, the (identical) scores of the candidates in $\mathcal{P}_k$ are strictly less than those of any $c \in \mathcal{W}_k$, and the manipulator's expected utility from $\mathcal{P}_k$ is maximized. Notice that the requirement $|\mathcal{P}_k| > \ell - k$ is necessary; otherwise, $\mathcal{P}_k \cup \mathcal{W}_k$ are the confirmed winners, which contradicts our objective of having $\mathcal{W}_k$ as the confirmed winners. We then compute the manipulator's expected utility from having the candidates in $\mathcal{W}_k$ as the confirmed winners and the candidates in $\mathcal{P}_k$ as the pending winners, and select a triple $(k, \mathcal{W}_k, \mathcal{P}_k)$ that maximizes the manipulator's expected utility.

The candidate set C has at most $\sum_{k=1}^{\ell} \binom{m}{k} \in \mathcal{O}(m^\ell)$ subsets of size at most ℓ; thus, it remains to show that for each subset of size at most ℓ the procedure described in the previous paragraph can be implemented in polynomial time. Fix a $k \leq \ell$ and a set $\mathcal{W}_k$. First, we pick k entries of α; these are the scores that we will assign to candidates in $\mathcal{W}_k$. There are $\binom{m}{k} = \mathcal{O}(m^\ell)$ ways of choosing such a set of scores; we go over all possible choices. We then order the candidates in $\mathcal{W}_k$ by decreasing order of scores under $\mathbf{s}$, and assign the lowest among the selected k scores to the first candidate, the second lowest to the second candidate and so on. If $\mathcal{W}_k$ can be made confirmed winners under some assignment of the k scores selected, then in particular they can be made confirmed winners under this assignment. Now, let $H_1, \dots, H_p$ be the levels of the candidates in $C \setminus \mathcal{W}_k$. We renumber the candidates in $C \setminus \mathcal{W}_k$ so that for all $i \in 1, \dots, p-1$, all candidates in H_i are before the candidates in H_{i+1}. Given a level H_i, we order the candidates in H_i so that if $c, c' \in H_i$ and the manipulator prefers c to c', then c' precedes c. Let $\alpha' = \{\alpha_{i_1}, \dots, \alpha_{i_{m-k}}\}$ be the remaining $m-k$ scores that the manipulator needs to assign; we assume $\alpha_{i_1} \leq \dots \leq \alpha_{i_{m-k}}$.

We assign $\alpha_{i_1}, \dots, \alpha_{i_{|H_1|}}$ to H_1 in that order. Similarly, we assign $\alpha_{i_{|H_1|+1}}, \dots, \alpha_{i_{|H_1|+|H_2|}}$ to H_2 and so on until all scores are assigned. This assignment, denoted σ_0, ensures that at each level, the manipulator's favorite candidates from that level receive the highest scores. Let Φ be the highest score of any candidate in $C \setminus \mathcal{W}_k$ under σ_0. Observe that for every score assignment to candidates in $C \setminus \mathcal{W}_k$ the score of some candidate in $C \setminus \mathcal{W}_k$ after the manipulator's vote is at least Φ. Thus, if Φ is greater than or equal to the score of some $c \in \mathcal{W}_k$, then $\mathcal{W}_k$ cannot be made confirmed winners using this score assignment, and we proceed to check a different assignment of scores to $\mathcal{W}_k$. Thus, from now one we assume that the score of each candidate in $\mathcal{W}_k$ is greater than Φ. Let $\mathcal{P}_0$ be the set of candidates whose score is Φ after submitting σ_0. We can try to modify σ_0 in order to increase the manipulator's utility, by swapping some candidates in the vote. Note that reassigning scores given to members of $\mathcal{P}_0$ will either result in a non-tied outcome, or decrease the manipulator's expected utility from the set of tied candidates. Indeed, suppose that a candidate $c \in \mathcal{P}_0$ received a score of β and now receives a higher score β'; this increases his score to be strictly more than Φ. If this results in a strictly higher utility for the manipulator, this means that the manipulator can strictly increase his utility by greedily assigning the highest scores in α' to the candidates he prefers the most, with no ties formed. On the other hand, if we assign a lower score to c, this means that some other candidate in a higher level receives a higher score, and the same argument applies. Thus, any swap we make will only involve candidates not in $\mathcal{P}_0$. However, note that the manipulator's utility is unaffected by candidates whose score is less than Φ. Thus, for any candidate c not in $\mathcal{P}_0$, we can just check if there is some score that will give him a total score of Φ. If such $c \in (C \setminus \mathcal{W}_k) \setminus \mathcal{P}_0$ exists, and adding c to $\mathcal{P}_0$ increases the manipulator's expected utility, we can add c to $\mathcal{P}_0$. Having done so for each candidate, we denote the resulting set by $\mathcal{P}_1$. We claim that $\mathcal{P}_1$ is indeed the set of pending candidates we require. However, it is not guaranteed that $|\mathcal{P}_1| > \ell - k$. If it is, then we are done. Otherwise, there are two cases.

Case 1: Given $\mathcal{W}_k$ and the scores we assign $\mathcal{W}_k$, it is impossible to find a score assignment such that $\mathcal{W}_k$ are confirmed winners.

Case 2: Even if there is a set $\mathcal{P}_k$ of pending winners, there is a set $\mathcal{P}'$ of candidates of size exactly $\ell - k$ such that the manipulator's utility from $\mathcal{W}_k \cup \mathcal{P}'$ is at least his expected utility from having $\mathcal{W}$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners.

Observe that both cases imply that if $|\mathcal{P}_1| \leq \ell - k$ we can just move on to another score assignment to $\mathcal{W}_k$, and ignore the current assignment: it is either impossible to have $\mathcal{W}_k$ as the confirmed winners, or there is another candidate set with the same utility that can be made confirmed winners and will be found in some other iteration. We must show that indeed one of these two cases holds.

If neither case holds, there exists a vote σ' such that if the manipulator submits σ', the set of confirmed winners is $\mathcal{W}_k$, the set of pending winners is $\mathcal{P}_k$, and for any set $\mathcal{P}' \subseteq C \setminus \mathcal{W}_k$ such that $|\mathcal{P}'| = \ell - k$ and the set $\mathcal{W}_k \cup \mathcal{P}'$ is a feasible set of winners it holds that the manipulator's expected utility from having $\mathcal{W}$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners is greater than his utility from $\mathcal{W}_k \cup \mathcal{P}'$.

First, consider the case where both confirmed and pending candidates get a total score of more than Φ points. Let $c_{j_1}, \dots, c_{j_{\ell-k}}$ be the manipulator's most preferred $\ell - k$ candidates in $\mathcal{P}_k$; by assumption, we must have that the manipulator's expected utility from $\mathcal{P}'$ is at most $\sum_{p=1}^{\ell-k} u(c_{j_p})$. Let $\mathcal{S}$ be the set consisting of these $\ell - k$ candidates and $\mathcal{W}_k$. Consider any candidate $c_j \in \mathcal{S}$ and suppose the manipulator grants $\alpha_{j'}$ points to c_j. The score of c_j after the manipulator votes is strictly more than Φ; thus $j' < j$. We set $\mathcal{S}' = \{c_{j'} \in C \mid \alpha_{j'} \text{ is assigned to some } c_j \text{ under } \sigma'\}$.

Now, consider the vote obtained from σ_0 by swapping the votes given to c_j and its corresponding candidate $c_{j'}$. Observe that some candidates can be in two such swaps—once acting as c_j and once as $c_{j'}$—in this case we begin from the swap which uses the candidate as c_j and afterwards we use the candidate who was put on his place for the next swap. All candidates in $C \setminus \mathcal{S} \setminus \mathcal{S}^{'}$ do not have their scores changed, so they still get at most Φ points; more importantly, all candidates in $\mathcal{S}$ now get strictly more than Φ points. Further, all candidates in $\mathcal{S}' \setminus \mathcal{S}$ get less than Φ points. Thus, in this case $\mathcal{S}$ are the confirmed winners and the manipulator's expected utility is at least as high as that from having $\mathcal{W}_k$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners, a contradiction. The other case is when the candidates in $\mathcal{W}_k$ have more than Φ points, but the candidates in $\mathcal{P}_k$ have exactly Φ points. This case is handled similarly; we omit the details due to space constraints. $\square$

6 Conclusions

Implementing a randomized tie-breaking rule proves to be an interesting new direction in computational social choice. Some voting rules (such as scoring rules and Bucklin) remain manipulable when employing randomized tie-breaking; however, computational barriers to manipulation arise for Copeland and Maximin. We also show that the target committee size does not affect the complexity of manipulating k-Approval, and procedures for choosing a constant-size committee that are based on scoring rules are manipulable as well.

While the picture for the single winner case is fairly complete, some problems in the multi-winner case remain open. For example, it is unclear whether $\mathcal{F}_\alpha$-MULTIRANDMANIPULATION remains in P if the size of the committee is unbounded, apart from the special case shown here for k-Approval. Moreover, the effects of randomized tie-breaking on coalitional manipulation are also unclear. While the hardness results shown in our work immediately imply hardness for coalitional manipulation under the same voting rules, the easiness results do not easily generalize.

To conclude, tie-breaking rules strongly influence the manipulability of elections; even when they do not induce hardness of manipulation, the techniques required in order to manipulate under randomized tie-breaking are quite different from those employed for lexicographic tie-breaking. This suggests that the choice of a tie-breaking rule is an important aspect of designing a good voting system and should not be ignored.

	P	NP-hard
Single-Winner ($\ell = 1$)	Plurality w/Runoff Maximin (restricted) Simplified Bucklin Classic Bucklin	Copeland Maximin (general) STV Ranked Pairs
Multi-Winner ($\ell \geq 1$)	Scoring Rules (for constant ℓ) k-Approval	

Table 1: Complexity of RANDMANIPULATION and MULTIRANDMANIPULATION for classic voting rules.

References

[1] J. J. Bartholdi and J. B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[2] J. J. Bartholdi, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[3] J. R. Chamberlin and P. N. Courant. Representative deliberations and representative decisions: Proportional representation and the borda rule. *The American Political Science Review*, 77(3):718–733, 1983.

[4] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *IJCAI'09*, pages 109–115, 2009.

[5] Y. Desmedt and E. Elkind. Equilibria of plurality voting with abstentions. In *ACM EC'10*, pages 347–356, 2010.

[6] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.

[7] P. Faliszewski and A. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31:53–64, 2010.

[8] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[9] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[10] A. Gibbard. Manipulation of schemes that mix voting and chance. *Econometrica*, 45:665–681, 1977.

[11] N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. Evaluation of election outcomes under uncertainty. In *AAMAS'08*, pages 959–966, 2008.

[12] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, September 2008.

[13] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *IJCAI'11*, pages 319–324, 2011.

[14] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *AAMAS'11*, pages 71–79, 2011.

[15] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[16] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer, and J. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *IJCAI'09*, pages 348–352, 2009.

[17] M. Zuckerman and J. Rosenschein. Manipulation with randomized tie-breaking under Maximin (extended abstract). In *AAMAS'12*, pages 1315–1317, 2012.

Svetlana Obraztsova, Yair Zick, Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
Email: `svet0001,yair0001@e.ntu.edu.sg;eelkind@ntu.edu.sg`

Coordination via Polling in Plurality Voting Games under Inertia

Reyhaneh Reyhani and Mark C. Wilson and Javad Khazaei

Abstract

We discuss a new model for strategic voting in plurality elections under uncertainty. In particular, we introduce the concept of *inertia* to capture players' uncertainty about poll accuracy. We use a sequence of pre-election polls as a source of partial information. Under some behavioural assumptions, we show how this sequence can help agents to coordinate on an equilibrium outcome. We study the model analytically under some special distributions of inertia, and present some simulation results for more general distributions. Some special cases of our model yield a voting rule closely related to the Instant Runoff voting rule and give insight into the political science principle known as Duverger's law. Our results show that the type of equilibrium and the speed of convergence to equilibrium depend strongly on the distribution of inertia and the preferences of agents.

1 Introduction

Voting as a preference aggregation method is widely used in human society and artificially designed systems of software agents. A large amount of recent research has considered the situation where a single individual or a small coalition attempts to manipulate an election result in its favour, assuming the remaining agents are naive (that is, always vote sincerely). Such an assumption on agent behaviour can be justified if the goal is to prove computational hardness results. However, if we wish to understand how voting rules function under fully strategic behaviour, we need to study a game-theoretic model of strategic manipulation.

The plurality rule is the most widely used voting rule, despite substantial criticism from social choice theorists. One point in its favour is its simplicity and space-efficiency: an agent needs only report a single alternative instead of submitting a full preference order, a list of utilities, or a binary approval vector, as is the case with most other rules. However, even such a simple rule can become complicated when strategic voting behaviour is considered. In this paper, we study plurality voting under the assumption that all agents act strategically, as a starting point for a study of further classes of rules.

Voting games notoriously have many equilibria, and agents often cannot coordinate on a particular equilibrium outcome. Hence, voting games are hard to understand. The lack of publicly known information can exacerbate the lack of coordination of agents. A commonly used device that addresses the coordination issue, especially for plurality elections, is to use publicly announced pre-election polls. Such polls, which amount to an approximate simulation of an election with the same agents and alternatives, increase the commonly known information among agents and may influence their strategic behaviour. However, the beliefs of agents regarding the accuracy of these results can be different. This is a key point in the present paper, and we introduce the concept of inertia to describe these differences in beliefs.

Several authors from the political science and economics disciplines have discussed the influence of pre-election polls in plurality elections, both empirically and theoretically. The key topic of interest is what is called "Duverger's law", a general political science principle stating that plurality voting tends to lead to two-party competition [13]. More recently some papers have appeared that study equilibria in plurality voting games from a more algorithmic viewpoint (e.g. [6], [1]). Most of the models that have been used, with a few exceptions (e.g. [3], [6]), concern static equilibria, classifying them as “duvergerian” or “non-duvergerian”, and fail to discuss the dynamic process of converging to equilibria via the use of polls. There are several important differences between our work and existing literature. One of the differences is related to the different amount of information and strategic behaviour of agents. The other extra feature considered in the present paper is agent-dependent beliefs about the reliability of this information.

1.1 Our Contribution

We present a model for plurality elections that allows for heterogeneous agents. We introduce the concept of an agent’s *inertia*, which is that agent’s perception of the accuracy of the poll result. This perception is the result of each agent’s belief about such sources of error as coverage bias, miscounting, roundoff error, and noise in the announcement of results. This concept is rather general and seems realistic enough to be used for both human society and for designed systems of autonomous agents. This article focuses on the plurality rule, places some restrictions on agent behaviour, and considers some particular distributions of inertia. We present some numerical and analytic results on convergence to equilibria, both duvergerian and non-duvergerian. For example, a duvergerian equilibrium often occurs when all agents have the same value of inertia.

2 Game Model

We have a set of agents each of whose set of allowable actions is to vote for a single alternative (not necessarily their most desirable alternative). Abstention is not allowed. Each agent has a total order on the set of alternatives (indifference is not allowed) but as the voting rule is plurality, they vote for one alternative. Agents participate in a sequence of pre-election polls before the real election. In our model, these polls include all agents and alternatives in real election, not just a random sample. The information that these polls reveal does not have any effect on the agents’ sincere preference order. In fact, we are interested in the strategic voting effect of polls rather than the so-called bandwagon or underdog effects considered in some papers [5]. In those papers, agents do not have a fixed preference order and their preference for an alternative is influenced by the popularity of that alternative.

We now discuss the assumptions in our model regarding the information and strategic behaviour of agents.

The information available for agents

The amount of information available to agents is a very important factor in their choice of strategy. The effect of poll information on the election result has been discussed in [12]. Complete information in plurality voting has been assumed in [8] and there is incomplete information in [11].

In the context of a repeated game, such as this sequence of polls under the plurality rule, in order to have complete information each agent would have to know how many agents of each *type* (sincere preference order) there are (this is usually called the *voting situation*). Even if this is unknown, we might expect to know the number of agents expressing each preference order in the previous poll. However, opinion polls for plurality will typically report only the number of agents ranking each alternative first, which we call the *scoreboard*. This lack of information on further preferences of other agents is crucial in the analysis below.

We use the concept of *inertia* to describe the reaction of agents toward the announced poll result. Agent coverage bias, miscounting or error and noise in announcing the result cause different values of uncertainty. This uncertainty brings about an inertia in agents. Each agent has an inertia value from the interval $[0, 1]$. An agent with inertia value of zero believes that the poll result is accurate. However, the poll result is meaningless to an agent with inertia value of one. In fact this agent does not consider the poll result in his decision making process. Other agents lie between these two extremes. Each agent's inertia value does not change during the sequence of polls. This seems reasonable because the set of participants in each poll does not change (it is always the entire set of agents), and the same system is used for counting and announcing the results in polls.

As far as we know this concept is new. The probability of miscounting has been discussed in [8], but is the same for all agents, whereas we have different values of inertia for different agents. The Poisson model of population uncertainty, in which there is uncertainty about the numbers of each type of agent, has been considered in [10]. In this paper agents have beliefs about these numbers that have been modelled as independent Poisson random variables. However, in our model, each agent just knows his own inertia and sincere preference order, and the scoreboard after each poll. This assumption makes sense for a system with no communication or coordination. This incomplete information influences the equilibrium result. Roughly speaking, it allows more alternatives to remain viable from the viewpoint of each agent.

The strategic behaviour of agents

The voting game described so far is still very general and allows for a wide range of outcomes. Voting games with more than two alternatives have many Nash equilibria and are not necessarily dominance solvable [2]. Eliminating dominated strategies is not sufficient to determine the result. Other refinements of equilibria such as strong and coalition-proof Nash equilibria do not always exist [7]. Some authors try to restrict the strategies of players by additional assumptions such as by assuming no voting for an alternative from another party [9].

In this paper, we assume agents have lexicographic preferences. Each agent infinitely prefers alternative x to alternative y, so he does not ignore any chance of winning of a more preferred alternative x [4]. Lexicographic preferences are not consistent with the idea of a cardinal utility function and probabilities are not relevant. Rather, they give a strong bias toward sincere voting which can still be overcome when an alternative is perceived to be a definite loser.

We also assume that each voter votes in each poll in the same way that he would if that poll were the actual election. One scenario in which this would occur is when voters do not know whether the current poll is the actual election. For example, the system designer may introduce this requirement. Thus voters will not attempt to vote strategically in the sense of misleading other voters, although they do vote strategically in the sense of playing their perceived best response. Note that the restricted information given by the scoreboard helps in this regard. For example, if bca voters could infer how many cab voters there were , they could vote for c in order that the cab voters do not abandon c, which might allow a to defeat b.

Therefore, agents vote for their most preferred alternative whom they perceive as having a non-zero chance of winning in further polls.

After each poll, each agent considers a set W of potential winners, consisting of all alternatives whom that agent perceives as having non-zero chance to win sometime in future. This set does not depend on the agents' preference order and only depends on the scoreboard and his inertia value. Agents update this set after the announced result of each poll. Agents start by voting sincerely in the first poll. Then, they update their votes according to their beliefs about potential winners during the sequence of polls. All these assumptions on behaviour are common knowledge as far as agents are concerned.

3 Game Dynamics

3.1 Notation

There is a set C of alternatives (we use index c for alternatives) which has m members, and a set V of players with n members (we use index ν for agents). We consider a sequence of K polls indexed by k, where the last poll is the election. However, agents are not aware of the value of K. Each agent has a sincere strict preference order on alternatives. There are $m!$ different preference orders (or types) which are indexed by t. We have plurality as our scoring rule in which each agent votes for only one alternative. Therefore, we can assume that the set of possible strategies for player ν is $S_\nu = C$. We use the following notations through the paper:

- $s_k(c)$: the normalized score of alternative c in poll k, namely the proportion of agents who have voted for c at poll k,
- $c_k(h)$: the alternative who has h-th highest score in poll k (e.g. $c_k(1)$ is the winner of poll k, note that we do not consider ties in this paper as this case occurs relatively rarely in large electorates),
- v_t: the number of agents with type (or preference order) t,
- $W_{\varepsilon,k}$: the set of potential winners from the view point of player with inertia value ϵ according to the result of poll k,
- $V_{c,k}$: the set of agents who vote for alternative c in poll k.

Definition 1 (The concept of certain and doubtful). Suppose that according to the poll result $s_k(i) < s_k(j)$. An agent with inertia ε is *certain* about this statement if

$$(1+\varepsilon)s_k(i) < (1-\varepsilon)s_k(j). \quad (1)$$

Otherwise, he is *doubtful*.

Note that this formula implies that if inertia of an agent is 0, then he will always be certain that j is ahead of i provided that such a result is reported. Also, Equation (1) implies that an agent with inertia equal to 1 will always be doubtful of any claimed scores.

The supporters of each alternative may be certain that the score of their favoured alternative is less than the winner, yet they might still consider that alternative as a potential winner and vote for him in the next poll. We study the concept of potential winner in the next section.

Example 1. Consider a 3 alternative election, and suppose the result of poll k is $s_k(c_k(1)) = 45\%$, $s_k(c_k(2)) = 30\%$ and $s_k(c_k(3)) = 25\%$. Any agent with inertia less than $\frac{1}{11}$ is certain that alternative 3 has fewer votes than alternative 2, but agents with inertia more than that are doubtful about this statement. In other words, those with $\varepsilon > \frac{1}{11}$ do not use this statement, while the others consider it in their strategic computations.

3.2 Set of potential winners

In the initial state ($k = 0$), an agent with inertia ε does not have any information about the number of supporters of each alternative. Therefore, he sees all alternatives as potential winners, $W_{\varepsilon,0} = C$, and he votes sincerely in the first poll. For the next poll, the agent votes for the most desirable alternative who can win in future (not necessarily the next poll) according to his interpretation of the poll result and the voting strategies of other agents (the strategy of agents is common knowledge).

Each agent's set of potential winners should satisfy some basic properties. The key necessary properties that we require are as follows. These are all common knowledge.

- non-emptiness: Any agent with any inertia value ε believes that there exists at least one agent with a positive chance of winning. W should clearly be nonempty for every voter, and contain the highest scoring candidate in the current poll.

- upward closure: if an agent with inertia ε believes that $c_k(x) \in W_{\varepsilon,k}$, then he believes $c_k(x-1) \in W_{\varepsilon,k}$. This seems reasonable: if an agent believes that some alternatives have a chance to win in future in the best case, then that agent also believes that all alternatives with higher current poll support also have a chance to win in future.

- overtaking: a possible winner must be able to overtake a higher scoring candidate who is also a possible winner. Overtaking the next higher scoring alternative is a necessary condition for winning, because the only chance an alternative has for attracting more support is that he improves his ranking position in the scoreboard. This is justified by the belief of agents about the upper closure of set of potential winners. For overtaking, alternative $c_k(x)$ needs extra support, and this support can only be obtained from the supporters of alternatives with a lower score than alternative $c_k(x)$. This is because agents who have already voted for higher scoring alternatives than $c_k(x)$ will change their votes to $c_k(x)$ if they perceive that their current choice does not have any chance to win. Upper closure of $W_{\varepsilon,k}$ would then lead to inconsistent beliefs.

  If $c_k(x)$ cannot overtake $c_k(x-1)$ in the next poll, in the most favourable case, then $x \notin W_{\varepsilon,k}$. We describe this case precisely in Proposition 1.

We first give an example to give the intuition behind our definitions.

Example 2. Consider scoreboard $(a, b, c, d) = (40\%, 29\%, 21\%, 10\%)$ and agent ν with $\varepsilon = 0$. Voter ν reasons as follows: for each agent with inertia ε, either alternative $d \in W_{\varepsilon,k}$ or not. If yes, then also alternatives $a, b, c \in W_{\sigma,k}$ (upward closure). The agents whose most desirable potential winner is alternative d have already voted for him, and the other agents prefer to vote for alternatives a, b or c in the next poll. Thus, the score of d cannot be increased and $d \notin W_{0,k}$. However, alternative $c \in W_{0,k}$ because it is possible that all supporters of alternative d switch to c, yielding scoreboard $(40\%, 29\%, 31\%, 0)$, and c can overtake alternative b, and in the next round all b-supporters may switch to alternative c, and he can overtake alternative a. Because of upward closure $b, a \in W_{0,k}$.

The basic properties above show that the currently highest-scoring alternative is always considered a potential winner by each agent. The necessary conditions do not define W uniquely. Because of lexicographic preferences, voters do not abandon candidates easily, and so it makes sense that W should be as large as possible. Of course if voters voted differently in the polls and the election (for example if they know that the next round is the election and have no other constraints on strategic action), W might be smaller. For example, a candidate may be able to win by successively attracting support from others, but the number of rounds remaining may not be enough for this to occur. We are ruling out this case by our assumptions on voter behaviour. For example, uncertainty about the time of the actual election allied to lexicographic preferences implies that W should be as large as possible. Thus we argue that the necessary conditions are sufficient.

We now show how to define the set of potential winners recursively starting from the top scoring alternative.

Definition 2. For $2 \leq i \leq m$, define condition $C_{ik\varepsilon}$ by

$$(1+\varepsilon)\sum_{h \geq i} s_k(c_k(h)) > (1-\varepsilon)s_k(c_k(i-1)). \qquad (C_{ik\varepsilon})$$

Proposition 1 (The conditions for being a potential winner). *After the announced result of poll k, $c_k(x) \in W_{\varepsilon,k}$ if and only if all conditions $C_{ik\varepsilon}$ for $2 \leq i \leq x$ hold. Algorithm 1 computes the set $W_{\varepsilon,k}$.*

Proof. Upward closure shows that the best chance of $c_k(x)$ overtaking $c_k(x-1)$ consists in attracting all supporters of agents currently voting for alternatives $c_k(h)$ with $h > x$, and retaining

Algorithm 1 Function for constructing $W_{\varepsilon,k}$

Require: $k \geq 1$
  $W_{\varepsilon,k} = \{c_k(1)\}$
  for $i = 2$ to m **do**
    if Condition $C_{ik\varepsilon}$ holds **then**
      $W_{\varepsilon,k} = W_{\varepsilon,k} \cup \{c_k(i)\}$
    else
      break
    end if
  end for

all current supporters. This yields condition $C_{xk\varepsilon}$, and so Algorithm 1 is clearly correct. Since overtaking of even higher alternatives must occur also, unrolling the loop in Algorithm 1 yields the result. ☐

Remark 1. In the majority case from the viewpoint of an agent with inertia value ε, in which

$$(1-\varepsilon)s_k(c_k(1)) > (1+\varepsilon)\sum_{c \neq c_{k(1)}} s_k(c),$$

alternative $c_k(2)$ and consequently all other alternatives except $c_k(1)$ do not have any chance to win in the future. Thus, $W_{\varepsilon,k} = \{c_k(1)\}$.

Example 3. Suppose the result of poll k is $s_k(a) = 55\%$, $s_k(b) = 30\%$ and $s_k(c) = 15\%$. According to Proposition 1,

$$W_{\varepsilon,k} = \begin{cases} \{a\} & 0 \leq \varepsilon \leq \frac{1}{11}; \\ \{a,b\} & \frac{1}{11} < \varepsilon \leq \frac{1}{3}; \\ \{a,b,c\} & \frac{1}{3} < \varepsilon \leq 1. \end{cases}$$

Therefore, we have 3 different sets for $W_{\varepsilon,k}$ based on the inertia value of agents. In the first inertia value interval, agents perceive the result of poll k as a majority case. Therefore, their set of potential winners is a singleton and they vote for a in poll $k+1$. In the second inertia value interval, they vote for a or b in poll $k+1$ based on their preference order. For example, an agent with preference order cab votes for a and an agent with preference order cba votes for b in poll $k+1$. In the third case where agents have high inertia, they do not care about the announced result of the poll. In fact, they believe each candidate to be viable and they just vote sincerely in poll $k+1$. An agent with inertia value of 1 always votes sincerely, regardless of the poll result.

4 Equilibrium Results for some special cases

4.1 Zero inertia

In the special case where inertia is identically zero for all agents, the set of potential winners is identical for all agents. We show that in this case the sequence of polls converges to a duvergerian equilibrium, i.e., a two party competition. Note that the inertia value is fixed in all polls and also we assume there is no majority case.

Theorem 1 (**duvergerian equilibrium**). *In a plurality voting game with common inertia value* $\varepsilon = 0$*, the polling sequence yields a duvergerian equilibrium in a non-majority case after at most* $m-2$ *polls.*

Proof. Let m be the number of alternatives and $\varepsilon = 0$. As agents have the same value of inertia, either all agents perceive the result as majority case or all of them perceive it as a non-majority case. As we explained before, in the majority case, agents vote for the highest scoring alternative (refer to Remark 1). In a non-majority case, we have $(s_k(c_k(1)) \leq \sum_{c \neq c_k(1)} s_k(c)$. According to Proposition 1, $c_k(2) \in W_{0,k}$, therefore, $\mid W_{0,k} \mid \geq 2$.

For all $\nu \in V_{c,k}$ for which $c \in C \setminus W_{0,k}$, ν changes his vote to his most desirable alternative in $W_{0,k}$. Thus, $s_{k+1}(c) = 0$, for each $c \in C \setminus W_{0,k}$. According to Proposition 1, $c_k(m) \notin W_{0,k}$. Therefore, in each poll, at least the last scored alternative is eliminated and after at most $m-2$ polls, we have a duvergerian equilibrium. □

Remark 2. There is a connection with the voting method instant runoff (IRV). When $m = 3$, if inertia is identically zero then our assumptions mean that the plurality election is actually just IRV. For general inertia and general m, we could fix some $\beta > 0$ and require that the election system automatically deletes the alternative whose support becomes less than β for the next poll. If we assume that 2 alternatives do not reach this boundary β simultaneously, we again simulate IRV. However, our procedure is more general, as several alternatives may be eliminated at one step.

4.2 Constant non-zero inertia

Suppose that all agents have the same value of inertia θ, with $0 < \theta \leq 1$. Again note that the set of potential winners is identical for all agents at all times and the inertia value is fixed in all polls. This case is similar to the setup of Messner and Polborn [8] where the probability of miscounting is positive but small. Messner and Polborn introduce the concept of robust equilibrium and show that for plurality games with 3 alternatives, all such equilibria are duvergerian. However, in that paper, the value of θ is common knowledge between all agents, and this is not the case in our model. The behavioural assumptions of agents also differ. Paper [8] shows that duvergerian equilibrium happens in all robust equilibria of plurality games with 3 alternatives.

We consider a 3-alternative election with a large number of agents, with a fixed inertia value θ which is the same for all agents. W.l.o.g. we may assume that $s_1(c) < s_1(b) < s_1(a)$. We also assume there is no majority case (refer to Remark 1).

Proposition 2. *Let*

$$\theta' = \max\{\frac{s_1(a) - s_1(b) - s_1(c)}{s_1(a) + s_1(b) + s_1(c)}, \frac{s_1(b) - s_1(c)}{s_1(b) + s_1(c)}\}. \tag{2}$$

A c supporter with inertia $\theta \leq \theta'$ will change his vote to a or b in the second poll.

Proof. According to Proposition 1,

$$c \in W_{\theta,1} \Leftrightarrow \begin{cases} (1+\theta)(s_1(b) + s_1(c)) > (1-\theta)s_1(a) \\ (1+\theta)s_1(c) > (1-\theta)s_1(b) \end{cases}$$

Therefore, $c \in W_{\theta,1} \Leftrightarrow \theta > \theta'$, and $c \in C \setminus W_{\theta,1} \Leftrightarrow \theta \leq \theta'$. □

Theorem 2. *Consider a plurality voting game with $m = 3$, and fixed inertia value θ which is the same for all agents. Assuming a non-majority case, the polling sequence yields a duvergerian equilibrium after 1 poll if $\theta \leq \theta'$.*

Proof. Similar to previous case, as agents have the same value of inertia, either all agents perceive the result as majority case or all of them perceive it as a non-majority case. As we explained before, in the majority case, agents vote for the highest scoring alternative (refer to Remark 1). In a non-majority case, according to Proposition 2, as the inertia values of all agents are equal, c supporters abandon c immediately, and a duvergerian equilibrium is reached after one poll. □

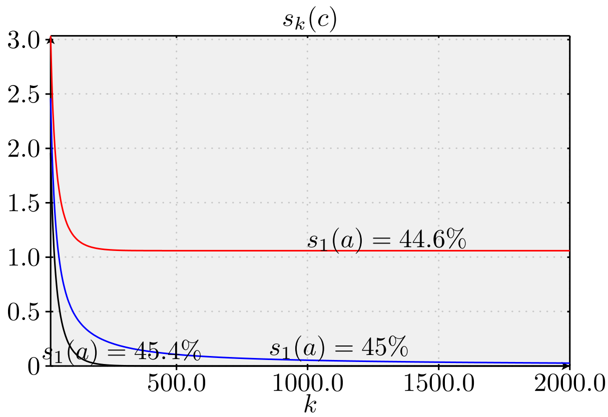


Figure 1: Score of the last alternative (c) as a function of k with uniform inertia distribution for three different cases where $V = (s_1(a), 35\%, 100\% - s_1(a) - 35\%, 5\%)$

Remark 3. Note that same constant non-zero inertia cases do not yield duvergerian equilibrium, depending on the value of θ. If $\theta > \theta'$, then every agent continues voting sincerely and the poll results will not change in the sequence.

Example 4. Consider plurality rule with 3 alternatives where the the scoreboard of the first poll is $(40\%, 35\%, 25\%)$. If the inertia value of all agents are θ and $\theta \leq \frac{1}{6}$, we have a duvergerian equilibrium.

4.3 Uniform distribution of inertia

We consider a 3-alternative election with a large number of agents, with a uniform inertia distribution on [0,1]. We describe the initial setup via a quadruple which is based on the first poll result $(s_1(a), s_1(b), s_1(c))$ and the true percentage v_6 of type cba agents (note this value is not known to any agent). W.l.o.g., we may assume that $s_1(c) < s_1(b) < s_1(a)$ and we approximate the discrete uniform distribution across agents by a continuous one for purposes of computation.

All c supporters who believe that c is a loser change their votes in favour of their second alternative. The percentage of type t agents (cab and cba) who vote in favour of alternative i (a and b respectively) in poll $k+1$ is denoted by $\alpha_{t,i,k}$. Note that the assumption of a common inertia distribution implies that for all k, $\alpha_{cab,a,k} = \alpha_{cba,b,k} \equiv \alpha_k$ and $\alpha_0 = 0$.

Proposition 3. *For a uniform distribution of inertia for all agents during the sequence of polls and initial result $V = (s_1(a), s_1(b), s_1(c), v_6)$, we have*

$$\alpha_k = \frac{1}{1 + \frac{2^k \left(\frac{s_1(c)-v_6}{s_1(b)+v_6}\right)^k (s_1(b)+v_6-2s_1(c))}{(s_1(b)-s_1(c))\left(-2^k\left(\frac{s_1(c)-v_6}{s_1(b)+v_6}\right)^k + \left(1-\frac{v_6}{s_1(c)}\right)^k\right)}} \tag{3}$$

Proof. According to the order of alternatives in the first poll and Proposition 1, a c supporter concludes that c is a loser and changes his vote if $(1+\varepsilon)s_k(c) < (1-\varepsilon)s_k(b)$.

Therefore, $\alpha_k = p\{\varepsilon < \frac{s_k(b)-s_k(c)}{s_k(b)+s_k(c)}\}$. The score of alternatives a, b and c in poll k is given by:

$$s_k(a) = s_1(a) + \alpha_{k-1}v_5 \qquad s_k(b) = s_1(b) + \alpha_{k-1}v_6 \tag{4}$$

$$s_k(c) = s_1(c) - \alpha_{k-1}v_6 - \alpha_{k-1}v_5 \tag{5}$$

Therefore,

$$\alpha_k = p\{\varepsilon < \frac{s_1(b) - s_1(c) + \alpha_{k-1}(s_1(c) + v_6)}{s_1(b) + s_1(b) - \alpha_{k-1}(s_1(c) - v_6)}\} \text{ for all } k \geq 1. \tag{6}$$

The stated solution formula for this recurrence is readily established by induction. □

Proposition 4. *The score of the last alternative in the first poll (which we denote by c) satisfies*

$$\lim_{k\to\infty} s_k(c) = \begin{cases} 0 & \text{if } s_1(b) + v_6 \geq 2s_1(c) \\ \left(\frac{2s_1(c) - v_6 - s_1(b)}{s_1(c) - v_6}\right) s_1(c) & \text{if } s_1(b) + v_6 < 2s_1(c) \end{cases} \tag{7}$$

Proof. The score of alternative c after $k+1$ polls is

$$s_{k+1}(c) = (1 - \alpha_k)s_1(c) \tag{8}$$

According to Proposition 3, if we converge k to infinity, we have

$$\lim_{k\to\infty} \alpha_k = \begin{cases} 1 & s_1(b) + v_6 \geq 2s_1(c); \\ \frac{s_1(b) - s_1(c)}{s_1(c) - v_6} & s_1(b) + v_6 < 2s_1(c). \end{cases}$$

The result follows immediately.

□

Remark 4. The convergence to zero is exponentially fast with the exponential rate decreasing as we approach the boundary between the two cases, and at the boundary it is subexponential. Figure 1 shows three special cases (the boundary case and 2 different cases in its neighbourhood).

Theorem 3. *In a plurality voting game with 3 alternatives and initial result $V = (s_1(a), s_1(b), s_1(c), v_6)$ and uniform distribution of inertia, the polling sequence yields a duvergerian equilibrium if and only if $s_1(b) + v_6 \geq 2s_1(c)$.*

Proof. Follows immediately from Proposition 4. □

Fig 1 illustrates this inequality when $v_6 = 5\%$ and $s_1(b) = 35\%$. For $s_1(a) \geq 45\%$, we have a duvergerian equilibrium.

4.4 Other distributions of inertia

The above results are for very special inertia distributions; explicit analysis of this type is not possible for general distributions. In this subsection, we investigate some different distributions via numerical simulations. Intuitively, we expect that distributions skewed to the left (with more agents of low inertia) will converge to the $\varepsilon \equiv 0$ case more quickly.

We consider the continuous triangular distribution $T(p)$ whose density function's graph has vertices at $(0,0)$, $(p,2)$ and $(1,0)$.

Example 5 (**The effect of inertia distribution: Triangular vs. Uniform**). Consider the initial result $V = (s_1(a), s_1(b), s_1(c), v_6) = (45\%, 35\%, 20\%, 5\%)$. According to Theorem 3, we have a limiting duvergerian equilibrium for uniform inertia distribution. Numerical results in Figure 1 (the line for $s_1(a) = 45\%$) also confirm this result. When we change the inertia distribution to be triangular with apex 0.5, we have the result in Figure 2. As we see in Figure 1, the convergence is very slow but changing the inertia distribution to $T(0.5)$ accelerates the process.

Example 6 (**The effect of voting situation**). In Figure 2, we have 5% cba agents. Figure 3 shows the result of the same situation with 10% cba agents which leads to a faster convergence. Note that the voting situation is not known to agents.

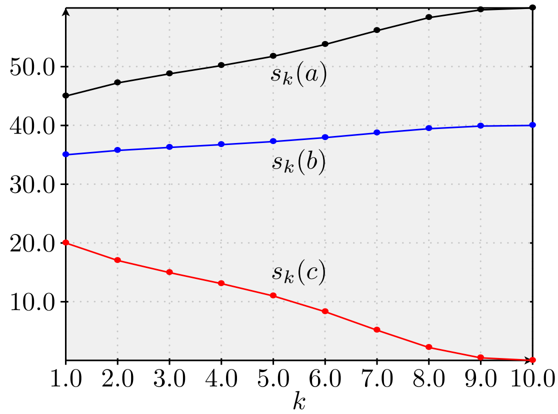


Figure 2: $V = (45\%, 35\%, 20\%, 5\%)$ and $T(0.5)$ inertia distribution

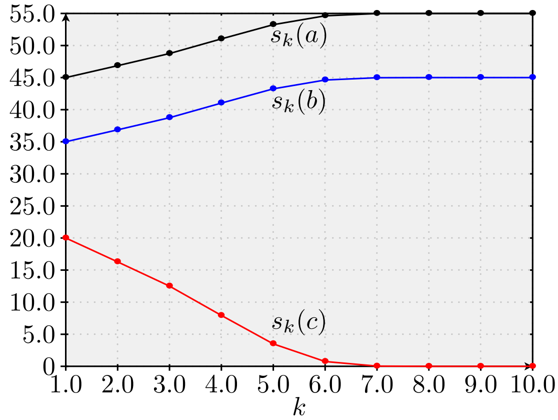


Figure 3: $V = (45\%, 35\%, 20\%, 10\%)$ and $T(0.5)$ inertia distribution

Example 7 (The effect of skewness of inertia distribution). Consider $V = (40\%, 35\%, 25\%, 10\%)$ with an inertia distribution of $T(0.5)$. This yields a non-duvergerian equilibrium, and it appears that the score of c converges to 22, as shown in Figure 4. However, the same voting situation with an inertia distribution $T(0.3)$ results in a duvergerian equilibrium as shown in Figure 5. In this case, more agents validate the poll result, and we have a duvergerian equilibrium after 10 polls.

5 Conclusion and Future Directions

In this paper we tried to study a repeated game with unknown number of rounds and incomplete information. The strategy of each player depends on his belief about the belief of other players. The sequence of opinion polls helps agents to coordinate on an equilibrium in an environment with some uncertainties about the accuracy of these polls. The amount of information available to agents has a critical role in influencing the strategic choices of agents. In this paper, we try to simplify the model with some assumptions about the strategy of players as a starting point for studying this game. Even in this simplified model, there are too many special cases that can happen depending on the inertia

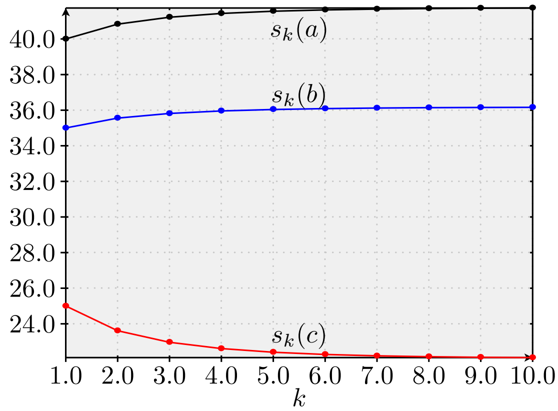


Figure 4: $V = (40\%, 35\%, 25\%, 10\%)$ and $T(0.5)$ inertia distribution

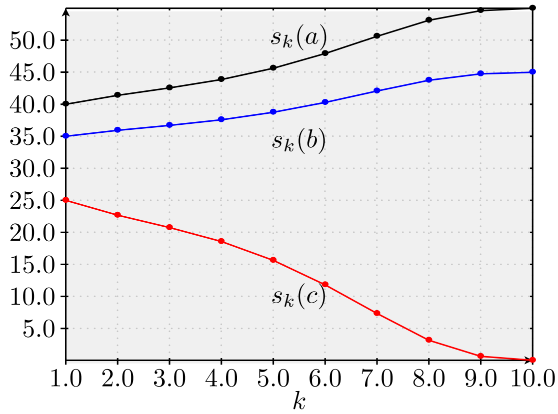


Figure 5: $V = (40\%, 35\%, 25\%, 10\%)$ and $T(0.3)$ inertia distribution

distribution or preference distribution of agents. We try to explain the model by some examples that give insight into different scenarios.

As a future direction, it is interesting to study how the strategy of agents will change if they have more information or in a more complicated model, each agent has different amounts of information. For example, some agents may have extra information than others regarding the inertia distribution of other agents or their preference order or the number of rounds ahead. Therefore, they may have different belief about the strategy of each agent.

Another interesting direction would be to to allow inertia to change from one poll to the next. For example, if random sampling is used instead of polling all voters, the sample size might vary between polls. More generally we want to explore the effect of inertia in other models with different behavioural assumptions for example, when voters use some some simple heuristic strategies. We expect to observe substantial differences in equilibrium outcomes when non-zero inertia is introduced into the model.

References

[1] Y. Desmedt and E. Elkind. Equilibria of plurality voting with abstentions. In *Proceedings of the 11th ACM conference on Electronic Commerce*, pages 347–356, 2010.

[2] A. Dhillon and B. Lockwood. When are plurality rule voting games dominance-solvable? *Games Econom. Behav.*, 46(1):55–75, 2004.

[3] M. Fey. Stability and coordination in Duverger's law: A formal model of preelection polls and strategic voting. *American Political Science Review*, 91(1):135–147, 1997.

[4] P.C. Fishburn. Axioms for lexicographic preferences. *The Review of Economic Studies*, 42(3):415–419, 1975.

[5] I. McAllister and D. T. Studlar. Bandwagon, underdog, or projection? opinion polls and electoral choice in Britain, 1979-1987. *The Journal of Politics*, 53(3):720–741, 1991.

[6] R. Meir, M. Polukarov, J.S. Rosenschein, and N.R. Jennings. Convergence to equilibria in plurality voting. In *Proceedings of AAAI*, volume 10, pages 823–828, 2010.

[7] M. Messner and M. K. Polborn. Strong and coalition-proof political equilibria under plurality and runoff rule. *International Journal of Game Theory*, 35:287–314, 2007.

[8] M. Messner and M.K. Polborn. Miscounts, Duverger's law and Duverger's hypothesis. Available at: http://works.bepress.com/polborn/25, 2011.

[9] D. P. Myatt and S. D. Fisher. Everything is uncertain and uncertainty is everything: Strategic voting in simple plurality elections. Economics Series Working Papers 115, University of Oxford, Department of Economics, 2002.

[10] R. B. Myerson. Comparison of scoring rules in Poisson voting games. *Journal of Economic Theory*, 103(1):219–251, March 2002.

[11] T.R. Palfrey. A mathematical proof of Duverger's law. In Peter C. Ordeshook, editor, *Models of strategic choice in politics*, pages 69–92. Ann Arbor: University of Michigan Press, 1989.

[12] A. Reijngoud and U. Endriss. Voter response to iterated poll information. *Proceedings of AAMAS2012, to appear*, 2012.

[13] W. H. Riker. The two-party system and Duverger's law: An essay on the history of political science. *The American Political Science Review*, 76(4):753–766, 1982.

Reyhaneh Reyhani
Department of Computer Science
The University of Auckland
Auckland, New Zealand
Email: `rrey015@aucklanduni.ac.nz`

Mark C. Wilson
Department of Computer Scince
The University of Auckland
Auckland, New Zealand
Email: `mcw@cs.auckland.ac.nz`

Javad Khazaei
Department of Engineering Science
The University of Auckland
Auckland, New Zealand
Email: `j.khazaei@auckland.ac.nz`

Empathetic Social Choice on Social Networks

Amirali Salehi-Abari and Craig Boutilier

Abstract

Social and economic networks play a fundamental role in facilitating interactions and behaviors between individuals, businesses, and organizations. It is widely recognized that such networks can correlate behaviors (and arguably preferences) among connected agents. We introduce a model for social choice—specifically, consensus decision making—on such networks that reflects certain interdependencies among agent utilities. Specifically, we define an *empathetic social choice framework* in which agents derive utility based on both their own intrinsic preferences and the satisfaction of their neighbors. We show how this problem translates into a weighted form of classical preference aggregation (e.g., social welfare maximization or certain forms of voting), and develop effective algorithms for consensus decision making that we believe should scale to large-scale (online) social or economic networks. Preliminary experiments validate the effectiveness of our proposed algorithms.

1 Introduction

Social networks play a central role in individual interactions and decision making. Indeed, it is widely acknowledged that the behaviors [7], and to a lesser extent the preferences, of individuals connected in a social network are correlated in ways that can be explained, in part, by network structure [10, 13]. Because of this, and the increasing availability of data that allows one to infer such relationships, the study of social choice problems on social networks is one of tremendous practical import. In fact, arguably most group decision problems, whether social, corporate, or policy-oriented, involve people at least some of whom are linked via myriad social ties. However, social choice in the context of social networks is something that has received, until recently, relatively little attention. Recent work has examined, for example, the formation of (hedonic) coalitions on social networks [6, 5], and stable matching on social networks [3, 16], in which the network captures one's affinity for potential partners. The influence of social networks on voting behavior has received considerable attention in the social sciences (e.g., [1, 14, 15]), and the emergence of online social network has even spawned computational research on the mechanisms to support delegation of votes in an online network [4].

In this paper, we consider the problem of *consensus decision making on social networks*, for example, in the form of voting over some option space. Specifically, we consider the problem of selecting a single option from a set of alternatives, for some group connected by a social network—e.g., a local constituency electing a political representative, or colleagues selecting a venue for a corporate retreat. While individuals have, as usual, personal *intrinsic utility* over the option space, we also incorporate a novel form of *empathetic utility* on social networks: in our model, the utility (or satisfaction) of an individual with a winning alternative a is a function of both her intrinsic utility for a and her *empathetic utility* for the "happiness" of her neighbors in the network. This use of empathetic utility can be seen as reflecting recent findings that suggest a person's happiness is influenced by the happiness of others with whom they are connected [11].

We consider two varieties of empathetic preference. In the first, the *local empathetic model*, the utility of individual i for alternative a combines her intrinsic preference for a with the *intrinsic* preference of i's neighbors for a, where the weight given to the preference of any neighbor j depends on the strength of the relationship between i and j. For instance, in selecting a restaurant, i may be willing to sacrifice some of her own intrinsic preference for the chosen restaurant if her colleagues are happier with the cuisine, and she defers more strongly to her closest friends. In the second, *global empathetic model*, i's utility for a depends on her intrinsic preference and the *total utility* that her neighbors have for a (not just their intrinsic preference). In other words, she doesn't just want her neighbors to be satisfied with a, she wants them to have high utility, which depends on the utility of *their* neighbors, and so on. For example, in voting for a political candidate, i may have a mild preference for a over b, but if b is strongly preferred by not only her closest neighbors, but also by their neighbors and many others in the community, she might prefer to see b elected so she won't have to interact with grumpy neighbors for the next five years.

Our main contributions in this paper are to develop a model for preference aggregation (e.g., certain forms of voting) that select consensus alternatives in a way that is sensitive to both intrinsic and empathetic preferences. Of course, we don't expect voters to actually compute such combined preferences; indeed, they may not have direct knowledge of the preferences of their neighbors. Instead voters specify their preferences for options and for the satisfaction of their neighbors (the latter could be inferred or estimated directly from the social network in some settings). We then propose methods for computing optimal alternatives under both the local and the global models. The former, unsurprisingly, corresponds to a simple form of weighted preference aggregation or weighted voting in which each voter implicitly "delegates" a portion of her vote to her neighbors. The latter, because individual utilities are co-dependent—indeed, utility spreads throughout the network much like PageRank values—requires the solution of a linear system to determine the optimal (fixed-point) option for the group. We describe (mild) conditions under which a fixed point is guaranteed to exist, and show that it too results in a form of weighted voting, where the weights assigned to each voter's intrinsic preference is readily derived from the solution to this linear system. Experiments explore various properties of our model and algorithms.

2 Social Empathetic Model

We begin by outlining our basic social choice model, motivating two notions of empathetic preference on social networks, and then defining socially optimal outcomes within this model. We also briefly discuss related work.

2.1 The Social Choice Setting

Apart from empathetic preferences on a social network, which we specify below, the choice framework we adopt is standard. We assume a set of alternatives $\mathcal{A} = \{a_1, \ldots, a_m\}$ and a set of agents $\mathcal{N} = \{1, \ldots, n\}$. Each agent j has *intrinsic preferences* over $\mathcal{A}$ in the form of either a (strict) preference ranking $\succ_j^I$ or a utility function u_j^I. For ease of presentation, we describe preferences in terms of utility functions, but discuss below on how to interpret voting procedures within our model. For example, in our experiments we use simple utility functions based on rankings of alternatives and score-based voting rules (specifically, Borda and plurality) to define "utility" for alternatives.

Our goal is to select a single consensus alternative $a^* \in \mathcal{A}$ that implements some social choice function f relative to the preferences of $\mathcal{N}$. For example, if agents' utilities were dictated solely by intrinsic preference and f were (utilitarian) social welfare, we would select $a^* = \arg\max \sum_j u_j^I(a)$. If preferences were given by intrinsic preference rankings, f would typically be represented by some voting rule (e.g., plurality or Borda).[1]

[1]Our model below applies directly to more general social choice problems, such as assignment/segmentation problems with network externalities (where individuals may be assigned different al-

2.2 Empathetic Preference on Social Networks

We depart now from the typical social choice framework by considering *empathetic preferences*, in which the preferences of one agent are dependent on those of others. We consider the specific case in which these influences are induced by connections in a social network (though the notion of empathetic preference need not be confined to networks). We focus on agent utility functions rather than preference rankings, since these allow the straightforward expression of quantitative tradeoffs between intrinsic and empathetic preference.[2]

Before discussing additional motivation, we introduce our model and notation. We assume a directed weighted graph $G = (\mathcal{N}, E)$ over agents, with an edge jk indicating that j's utility is dependent (in a way to be specified below) on its neighbor k's preference, the strength of this dependence given by edge weight w_{jk}. Naturally j's utility will usually depend on its own intrinsic preferences, so loops jj will usually be present. We assume that $w_{jk} \geq 0$ for any edge jk, and that $\sum_k w_{jk} = 1$ for any j (though allowing variable weightings to reflect, say, weighted voting schemes is also possible). For convenience, we treat missing edges as if they had weight zero (and vice versa). Thus, we represent the graph with a weight matrix $\mathbf{W} = [w_{ij}]$. We generally think of these edges as corresponding directly to some relationship in a social network, or possibly induced from such relationships. See Fig. 1(a) for an illustration.

We take j's utility for a to be a linear combination of it's own intrinsic preference for a and the empathetic preference derived from each of its neighbors—recall that we consider pure consensus/single-winner voting scenarios in which a single option a is selected for all $j \in \mathcal{N}$—where network weights determine the relative importance of each.[3] Letting $e_{jk}(a)$ denote the *empathetic utility derived by j from k*, we define j's utility $u_j(a)$ to be

$$u_j(a) = w_{jj}u_j^I(a) + \sum_{k \neq j} w_{jk}e_{jk}(a).$$

The ratio of w_{jj} to $\sum_{k \neq j} w_{jk}$ captures the relative importance of intrinsic and empathetic utility to j.

We consider two ways in which to define empathetic preferences e_{jk}. In the *local empathetic model*, we simply define $e_{jk}(a) = u_k^I(a)$; in other words, j's utility for a is simply a linear combination of intrinsic utilities of j's neighbor (including it's own):

$$u_j(a) = \sum_k w_{jk}u_k^I(a). \tag{1}$$

This model captures the fact that an agent j is concerned about the "direct" preference of a neighbor k for alternative a; but the fact that k's utility may depend on k's *own* neighbors does not impact j. For instance, consider a family or a group of friends deciding on a movie (or restaurant or outing): the preferences of certain family members (e.g., parents) for a specific film may depend on the preferences of others (e.g., children, whom they want to be entertained by the choice of film).

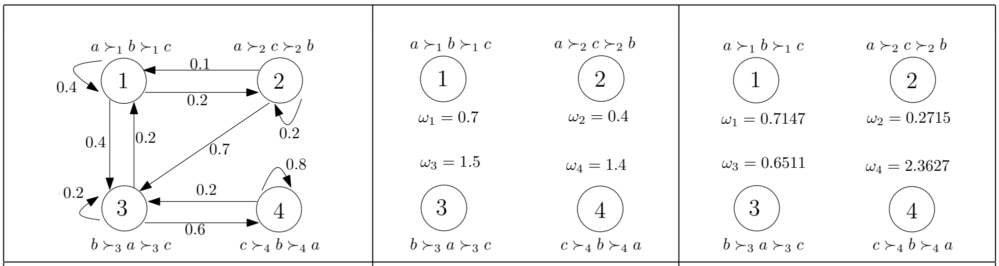
In the *global empathetic model*, we define $e_{jk}(a) = u_k(a)$, so that k's complete utility for a—which *may depend on k's own neighbors*—influences j's utility for a, giving rise to

$$u_j(a) = w_{jj}u_j^I(a) + \sum_{k \neq j} w_{jk}u_k(a). \tag{2}$$

ternatives), matching problems, and so on, without difficulty. Our algorithms are, however, specific to the "single-choice" assumption.

[2] Suitable *qualititative* expression of such tradeoffs is an important ongoing research direction.

[3] More general non-linear models are possible as well.



(a) social network with preferences | (b) local empathetic weights | (c) global empathetic weights

Figure 1: Social network and ranked preferences, with weights under the local and global empathetic model. Using Borda or plurality-based utility, the consensus winner is different in each model: a under intrinsic; b under local empathetic; c under global empathetic.

In this model, j's utility for a depends on the utility (not just intrinsic preference) of its neighbors for a. For example, a voter may care about the overall level of satisfaction of her neighbors when voting for a political representative, but recognize that there is a larger societal effect at work, where their satisfaction also depends on their neighbors, etc. More concretely, companies linked in complex supply chain may well care about the overall success of their suppliers and customers, and consider adopting industry-specific or economic policies in that light. In the global model, the circular dependence of utilities requires a fixed point solution to the linear system defined by Eq. 2 (see below).

Correlations of behavior and/or preferences among agents connected in social network is widely accepted, and can be explained by a variety of mechanisms [10, 13]. Among these are: *technology/information diffusion*, in which agents become aware of opportunities or innovations from connections to their neighbors; *network externalities*, in which the benefits of adopting some behavior increase with the number of neighbors doing the same; or *homophily*, in which people with similar characteristics (say, preferences) more readily form social ties. Our empathetic model is somewhat different in that a person's intrinsic preferences over options $\mathcal{A}$ are not presumed to be correlated with their neighbors, but their revealed preferences for $\mathcal{A}$ might be: their choices (or stated utilities) will generally reflect some consideration, however estimated, of their neighbors' preferences as well.

2.3 Social Welfare as Weighted Intrinsic Utilities

In realistic social choice situations, agents with empathetic preferences must often perform sophisticated reasoning not only about their intrinsic preferences for alternatives, but also about those of their neighbors. Thus, even in the local empathetic setting, expressing preferences (e.g., voting) is difficult since agents usually have incomplete (and in some cases, no) information about the preferences of their friends, neighbors, or colleagues. The global empathetic setting is even more complex, since an agent is further required to reason about her neighbors' network connections as well as their intrinsic/empathetic tradeoffs.

In our models, preference aggregation and optimization can be performed by simply having agents specify their *intrinsic preferences*, as is standard in social choice, and the *weights* they assign to neighbors in their local network. In social scenarios, this can remove a considerable informational and cognitive burden from agents who might otherwise be required to determine their total utility for alternatives. In other situations, agents might not wish to reveal their preferences to their neighbors, but might still want their neighbors to obtain a favorable result (consider, for example, a collection of companies, voting over some economic policy alternatives, that are linked together in complex supply chain relationships which correlates their stability or profitability). It turns out that, given a known network G, the problem of consensus decision making with empathetic preferences can be recast as a *weighted preference aggregation problem over intrinsic preferences alone.* Not only does this ease the burden on agents, it also allows one to recast the problem as one of simple weighted voting, or of weighted (utilitarian) social welfare maximization, rendering the decision making process itself fully transparent. Here we focus on social welfare maximization.

For the local model, determining the weights associated with each agents' intrinsic preference is straightforward. Assume network weights $\mathbf{W}$. Let $\mathbf{u}(a)$ be the n-vector of agent utilities to be computed as a function of the corresponding vector $\mathbf{u}^I(a)$ of intrinsic utilities for some fixed alternative a. By Eq. 1, we have $\mathbf{u}(a) = \mathbf{W}\mathbf{u}^I(a)$. Then letting $\boldsymbol{\omega} = \mathbf{e}^\top \mathbf{W}$ (where $\mathbf{e}$ is a vector of ones), the social welfare of any alternative a under the local empathetic model is given by

$$sw_l(a, \mathbf{u}^I) = \boldsymbol{\omega}^\top \mathbf{u}^I(a). \tag{3}$$

Thus social welfare maximization under local empathetic utility is simply weighted maximization of intrinsic preference, where the weight of j's intrinsic utility ω_j is simply the sum of the weights of its incoming edges.

Fig. 1(b) illustrates the local model in action. The derived weights for each agent are shown. We assume preference rankings, and suppose utilities are dervied from these using either Borda or plurality scores. We see that the decision can be different under the local model than using voting based on intrinsic preferences along (a wins in the intrinsic model, while b wins in the local model). Indeed, using score-based voting rules, we can readily interpret this model as a form of *empathetic voting*, where the weight one assigns to a neighbor can be interpreted as the extent to which one would sacrifice one's own preferences to improve a neighbor's intrinsic satisfaction with the winning alternative.

Things are slightly more subtle in the global empathetic model. Computing the utility vector $\mathbf{u}(a)$ for alternative a requires solving a linear system to compute the fixed point of Eq. 2. Unfortunately, a unique solution is not guaranteed to exist.[4] However, in addition to our assumptions above of *non-negativity* (i.e., $\mathbf{W} \geq \mathbf{0}$) and *normalization* (i.e., $\sum_k w_{jk} = 1$ for all j), a third mild condition on the social network (weight matrix $\mathbf{W}$) is sufficient to ensure a unique fixed point solution, namely, *positive self-loop*: $w_{jj} > 0$ for all j. Let $\mathbf{D}$ be the $n \times n$ diagonal matrix with $d_{jj} = w_{jj}$. We can write Eq. 2 as

$$\mathbf{u}(a) = (\mathbf{W} - \mathbf{D})\mathbf{u}(a) + \mathbf{D}\mathbf{u}^I(a). \tag{4}$$

As a consequence,

Theorem 2.1 (Fixed-point Utility) *Assuming nonnegativity, normalizaton, and positive self-loop, Eq. 4 has a unique fixed-point solution* $\mathbf{u}(a) = (\mathbf{I} - \mathbf{W} + \mathbf{D})^{-1}\mathbf{D}\mathbf{u}^I(a)$.

(Proofs of all results are included in the longer version of this paper.) As in the local model, social welfare maximization in the global model can be interpreted as weighted maximization of intrinsic preference (though with a less straightforward interpretation):

Corollary 2.1 *In the global empathetic model, social welfare of alternative a is given by* $sw(a, \mathbf{u}^I) = \boldsymbol{\omega}^\top \mathbf{u}^I$ *where* $\boldsymbol{\omega}^\top = \mathbf{e}^\top(\mathbf{I} - \mathbf{W} + \mathbf{D})^{-1}\mathbf{D}$.

Once again, in (score-rule based) voting contexts, one can interpret the global empathetic model as trading off one's own satisfaction with a winning alternative with the "overall" (not just intrinsic) satisfaction of one's neighbors: see Fig. 1(c) for an illustration. We discuss weight computation in Sec. 3.

[4]Consider two individuals j and k, with $w_{jj} = w_{kk} = 0$, $w_{jk} = w_{kj} = 1$, $u^I_j(a) = 0.1$, and $u^I_k(a) = 1$. The induced system does not have a unique fixed-point solution.

2.4 Related Work

We are unaware of other formal models which consider the dependency between agent utilities in a social network using the type of empathetic utility we introduce above. However, empathetic utilities might be viewed as a form of network externality in an agent's utility function, though unlike typical models of externalities, an agent's utility depends on the (latent) utility of its neighbors for the chosen alternative rather than the behavior of, or allocation made to (at least directly), her neighbors (or others).

Decision making in the presence of network externalities has recently attracted attention. Bodine-Baron *et al.* [3] study stable matchings (e.g., of students to residences) with *peer effects*: these local network externalities reflect the fact that students prefer to be assigned to the same residence as their friends in a social network. Brânzei and Larson address coalition formation on social networks in two different settings: (a) agent utility for a coalition depends on its affinity weights with others in the coalition [5]; and (b) agent utility depends on her distance to others on the induced social network [6]. The problem of auction design in social networks with positive network externalities is studied in [12].

Boldi *et al.* [4] consider voting on social networks, describing a form of *delegative democracy* in which an individual can either express her preferences directly, or to delegate her vote to a proxy from among her neighbors. In our model, individuals are not asked to delegate their votes or preferences: we simply consider the dependency of their preferences on those of others, though this can be viewed loosely as *implicit, partial* delegation of preferences.

3 Computing Winners in the Empathetic Models

We now consider the question of computing the social welfare maximizing alternative in both the local and global empathetic models. In Sec. 2.3, we observed that—for both the local and global empathetic models—social welfare can be expressed as $sw(a, \mathbf{u}^{\boldsymbol{I}}) = \boldsymbol{\omega}^\top \mathbf{u}^{\boldsymbol{I}}(a)$ for an appropriate weight vector $\boldsymbol{\omega}$. Given the vectors $\mathbf{u}^{\boldsymbol{I}}(a)$ for any $a \in \mathcal{A}$, we can readily compute the optimal alternative $a^* = \arg\max_{a\in A} \boldsymbol{\omega}^\top \mathbf{u}^{\boldsymbol{I}}(a)$, requiring $O(nm)$ time. Of course, this presupposes access to $\boldsymbol{\omega}$, which has different meanings in each model, and hence requires different approaches for its computation. In the global model, this suggests a different method for computing a^* as well, without (necessarily) requiring the full computation of $\boldsymbol{\omega}$.

We first consider the local model, where $\boldsymbol{\omega}^\top$ can be calculated easily with a single vector-matrix multiplication, $\boldsymbol{\omega}^\top = \mathbf{e}^\top \mathbf{W}$, in time $O(n^2)$. However, social networks are generally extremely sparse, with the number of outgoing edges associated with any node j in the graph bounded by some small constant c which is independent of the network size (generally, social networks, while potentially locally dense, are sparse in a global sense). In sparse networks, $\boldsymbol{\omega}$ can be computed much more efficiently: ω_j is simply the sum of j's outgoing edges weights. If the outgoing neighbors of any node are bounded by a constant, $\boldsymbol{\omega}$ can be computed in $O(n)$ time and a^* can be determined in the straightforward fashion mentioned above in $O(nm)$ time. Thus the complexity of computing optimal alternatives in the local empathetic model is no different than that of straightforward social welfare maximization of straightforward (e.g., scoring rule-based) voting.

In the global model, $\boldsymbol{\omega}^\top$ has a more complicated expression, $\boldsymbol{\omega}^\top = \mathbf{e}^\top \mathbf{A}^{-1}\mathbf{D}$ where $\mathbf{A} = \mathbf{I} - \mathbf{W} + \mathbf{D}$ (see Cor. 2.1). The difficulty lies largely in matrix inversion: $\mathbf{A}^{-1}$ can be computed via Gauss-Jordan elimination, which has complexity $O(n^3)$. This implies that straightforward computation of a^* requires $O(n^3 + nm)$ time. In general, matrix inversion is no harder than matrix multiplication (see, e.g., [9, Thm. 28.2]). Although efficient matrix multiplication is the topic of ongoing research (e.g., [8]), its complexity cannot be less than $O(n^2)$ since all n^2 entries must be computed. Therefore, straightforward computation of a^* in the global model cannot have complexity less than $O(n^2 + nm)$.

We expect n to be extremely large in at least some social choice problems on social networks, e.g., in the tens of thousands (number of people in a small town), the millions (large big cities), or hundreds of millions (large country, number of Facebook or Twitter users). This makes algorithms that scale more than linearly in n problematic, both from the perspective of time and memory. Of course, many iterative methods have been proposed for matrix inversion and solving linear systems (e.g., Jacobi, Gauss-Siedel) which have $O(n)$ complexity (in non-sparse systems) per iteration and tend to converge very quickly in practice. We now briefly describe the use of a standard Jacobi method for computing a^* in the global model. We first show how to compute the utility vector $\mathbf{u}(a)$ for each alternative a, and then propose an algorithm called *iterated candidate elimination (ICE)* that will compute the optimal a^* without (necessarily) computing each $\mathbf{u}(a)$ fully.

Consider first a simple iterative method for computing $\mathbf{u}(a)$. Let $\mathbf{u}^{(t)}(a)$ be the estimated utilities for alternative a after t iterations.

Theorem 3.1 *Consider the following iteration:*

$$\mathbf{u}^{(t+1)}(a) = (\mathbf{W} - \mathbf{D})\mathbf{u}^{(t)}(a) + \mathbf{D}\mathbf{u}^{I}(a).$$

Assuming nonnegativity, normalizaton, and positive self-loop, this method converges to $\mathbf{u}(a)$*, the solution to Eq. 4.*

For each $j \in \mathcal{N}$, the method computes:

$$u_j^{(t+1)}(a) = w_{jj}u^I(a) + \sum_{k \neq j} w_{jk}u_k^{(t)}(a). \quad (5)$$

We can interpret $u_j^{(t)}(a)$ as agent j's estimated utility for alternative a after t iterations. This updating scheme has a natural interpretation in terms of agent behavior: suppose that each individual is able to repeatedly observe her friends' revealed utilities, and updates her own utility for various alternatives in response. This process will eventually converge (this is true even if the updates are "asynchronous"). One can readily bound the error in the estimated utilities at the t^{th} iteration:

Theorem 3.2 *In the iterative scheme above,*

$$\left\|\mathbf{u}(a) - \mathbf{u}^{(t)}(a)\right\|_\infty \leq (1 - \tilde{w})^t \left\|\mathbf{u}(a) - \mathbf{u}^{(0)}(a)\right\|_\infty,$$

where $\tilde{w} = \min_{1 \leq i \leq n} w_{ii}$.

Hence, societies in which individuals have self-loops with relatively larger weight (i.e., less empathy) converge to fixed-point utilities faster societies with greater empathy (our empirical results below support this).

This error bound allows one to bound the error in estimated social welfare if the utilities of all alternatives are estimated in this fashion. Let $sw^{(t)}(a) = \sum_j u_j^{(t)}(a)$.

Theorem 3.3 *Assume* $u_j^I(a) \in [c, d]$ *and* $u_j^{(0)}(a) \in [c, d]$*, for all* $j \in \mathcal{N}$*. Under the conditions above, for any* t*:* $\left|sw(a) - sw^{(t)}(a)\right| \leq n(d - c)(1 - \tilde{w})^t$*, where* $\tilde{w} = \min_{1 \leq i \leq n} w_{ii}$.

As a result, we know that (under the same assumptions):

Proposition 3.4 *If* $sw^{(t)}(b) - sw^{(t)}(a) \geq 2n(d - c)(1 - \tilde{w})^t$ *then* $sw(b) > sw(a)$.

We can exploit Prop. 3.4 in a simple iterative algorithm for computing a^* we call *iterated candidate elimination (ICE)*. The intuition behind ICE is to iteratively update the estimated utilities of the subset $C \subset \mathcal{A}$ of candidates that are non-dominated, and gradually prune away any candidate that is dominated by another until only one, a^*, remains. Roughly, ICE first initializes $C = \mathcal{A}$ and $u_j^{(0)}(a) = c$ for all $j \in \mathcal{N}, a \in \mathcal{A}$. An iteration of ICE consists of: (1) updating estimated utilities using Eq. 5 for all j and $a \in C$; (2) computing estimated social welfare of each $a \in C$; (3) determining the maximum estimated social welfare $\hat{sw}^{(t)}$; (4) testing each $a \in C$ for domination, i.e., $\hat{sw}^{(t)} - sw^{(t)}(a) \geq 2n(d-c)(1-\tilde{w})^t$; and (5) eliminating all dominated candidates from C. The algorithm terminates when only one candidate (i.e., a^*) remains in C (the pseudo-code for the algorithm is provided in the longer version of this paper). The running time of ICE is at most $O(tmn^2)$ where t is the number of iterations required. More precisely, ICE runs in $O(tm|E|)$ time; and if the number of outgoing edges is bounded, $O(tmn)$. Our hope is that in practice, the methods converges in relatively few iterations, a fact indeed borne out in our preliminary experiments below. ICE also provides a natural means of approximation in large problems.

4 Empirical Results

We now describe some preliminary experiments on randomly generated networks and intrinsic preferences designed to test the differences in the decisions that result under standard non-empathetic, local empathetic and global empathetic models, the impact of these decisions on different agents, and the performance of the ICE algorithm.

Experimental Setup. Our test scenarios require generation of intrinsic preferences and a social network. We assume that individual intrinsic utilities arise from an underlying preference ordering over $\mathcal{A}$. In all experiments, we assume $m = 5$ or $m = 10$ alternatives, and draw a random preference ordering for each agent j under the impartial culture assumption (all permutations are equally likely). For simplicity, and to draw connections to voting on social networks, we assume j's utility is given by the Borda or plurality score of the alternative in its ranking. If treating these strictly as utility, they embody very different, extreme assumptions: Borda treats utility differences as smooth and linear, whereas plurality views utility in a more "all or nothing fashion."

We generate random social networks using a *preferential attachment* model for scale-free networks [2] (this is only one of many models that can be adopted). The model works in the following iterative fashion: start with n_0 initial nodes; we repeatedly add nodes (until we have a graph with n nodes), where each new node added is connected to $k \leq n_0$ existing nodes, and an existing node i is selected as a neighbor with probability $P_i = \frac{\deg(i)}{\sum_j \deg(j)}$. We set $n_0 = 2$ and $k = 1$ or $k = 2$ in all our experiments. We then convert the resulting undirected graph to an directed graph by replacing each undirected edge with the two corresponding directed edges; add a self-loop to each node with weight α; then add normalized weights to all other edges (all outgoing edges from j excluding the self loop have equal weights that sum to $1 - \alpha$). The parameter $\alpha \in (0, 1]$ represents the degree of self-interest, and $1 - \alpha$ the degree of empathy in the society.

Performance Metrics. To measure whether the different models result in difference decisions, we assume the agents *actual utility model* is one of intrinsic (non-empathetic), local or global. We then consider making decisions using any of these models as an *assumed utility model*, and measure the effect on actual utility (e.g., global empathetic utility) of making a decision using the assumed model (e.g., intrinsic). Since decisions might be different in each case, we measure the loss in social welfare due to making a decision using the incorrect model. Let $sw^{ac}(\cdot)$ and $sw^{as}(\cdot)$ be social welfare under the actual and assumed models, respectively, and a_w and a_s be the corresponding optimal alternatives (or winners). We

Actual Utility	Assumed Utility			WSWL
	intrinsic	**local**	**global**	
intrinsic	0%(0%)	1.45%(9.95%)	1.10%(8.00%)	5.59%(14.63%)
local	2.95%(19.28%)	0%(0%)	0.09%(3.21%)	11.22%(25.10%)
global	1.78%(12.73%)	0.07%(2.73%)	0%(0%)	9.01%(20.97%)

Table 1: Average (maximum) RSWL and WSWL: 2500 runs, Borda scoring, $m = 5$, $n = 1000$, $k = 1$, $\alpha = 0.25$.

Actual Utility	Assumed Utility		
	intrinsic	**local**	**global**
intrinsic	0.0%(0.0%)	28.4%(100.0%)	22.6%(100.0%)
local	28.5%(100.0%)	0.0%(0.0%)	1.2%(86.9%)
global	22.3%(100.0%)	1.1%(97.0%)	0.0%(0.0%)

Table 2: Average (maximum) NSWL: 2500 runs, Borda, $m = 5$, $n = 1000$, $k = 1$, $\alpha = 0.25$.

define *relative social welfare loss (RSWL)* to be $L(as, ac) = [sw^{ac}(a_w) - sw^{ac}(a_s)]/sw^{ac}(a_w)$ (we sometimes report it as a percentage). RSWL has a lower bound that is independent of the assumed model: let the alternative a^- have *minimum social welfare* under the actual model (so it is no better than the decision under the assumed model). *Worst-case social welfare loss (WSWL)* is defined as $W(ac) = [sw^{ac}(a_w) - sw^{ac}(a^-)]/sw^{ac}(a_w)$. Finally, it usually makes sense to normalize RSWL by considering the range of possible social welfare values actually attainable: *normalized social welfare loss (NSWL)* is simply $N(as, ac) = [sw^{ac}(a_w) - sw^{ac}(a_s)]/[sw^{ac}(a_w) - sw^{ac}(a^-)]$. This offers a more realistic picture of loss due to using an incorrect assumed utility model (by comparing it to the loss associated with making the *worst possible decision* under the actual model).

Social Welfare Loss. We first consider RSWL, WSWL and NSWL for all nine combinations of assumed and actual utility models. We fix $\alpha = 0.25$, $n = 1000$, $m = 5$, $k = 1$, and the scoring rule to Borda. We generate 50 random networks, and for each generate 50 intrinsic utility profiles (2500 problem instances), and compute RSWL and WSWL. Average (with maximum in parentheses) RSWL for various combinations of actual and assumed models is reported in Table 1 as are average (maximum) WSWL. Maximum RSWL is more than 19% and 12% when intrinsic utility is assumed but actual utility is local or global, respectively. Moreover, we can see that global vs. local and local vs. global are quite close. Notice that average differences are quite slight: this is because the impartial culture model, in essence, renders are alternatives quite close in terms of Borda or plurality score. By normalizing for the fact that most decisions are reasonably good, we get a more accurate picture of the loss incurred by using non-empathetic voting. NSWL is reported in Table 2, which shows that making the wrong assumptions can be quite damaging; e.g., the intrinsic model loses 22.3–28.5% of empathetic social welfare on average.

Since impartial culture is generally viewed as an unrealistic model of real-world preferences, we also tested our methods using preferences drawn from 2002 Irish electoral data from the Dublin West constituency, with 9 candidates and $29,989$ ballots of top-t form, of which 3800 are complete rankings.[5] Generating 1000-node networks as above, we randomly assign full rankings to nodes from this set of 3800 complete rankings. Results on RSWL and WSWL for plurality scoring are shown in Table 3. As above, average RWSL is slight; but the maximum values show significant social welfare loss in certain instances, especially when using the intrinsic model to make decisions for empathetic preferences.

Utility and Societal Weights. We now examine how individual utility—and its intrinsic and empathetic components—and computed weights depend on their degree of nodes in the social networks in global empathetic model. Using data from the previous experiment,

[5]See www.dublincountyreturningofficer.com.

Actual Utility	Assumed Utility			WSWL
	intrinsic	local	global	
intrinsic	0%(0%)	1.82%(33.62%)	1.30%(19.55%)	97.22%(99.62%)
local	2.64%(39.25%)	0%(0%)	0.01%(6.80%)	97.28%(99.85%)
global	1.53%(31.26%)	0.10%(8.4%)	0%(0%)	97.24%(99.77%)

Table 3: Average (maximum) RSWL and WSWL for West Dublin data set: 2500 runs, plurality scoring, $m = 9$, $n = 1000$, $k = 1$, $\alpha = 0.25$.

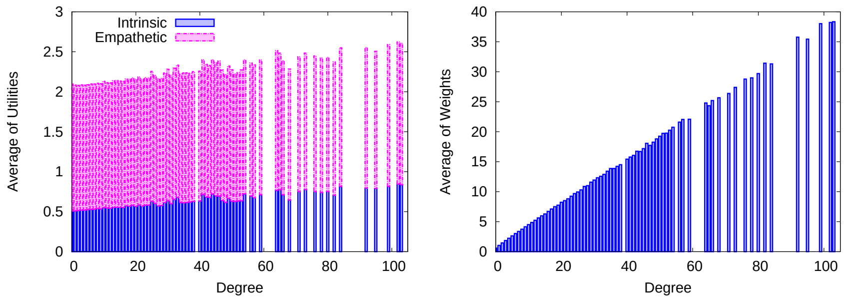


Figure 2: Average (a) instrinsic and empathetic utilities and (b) individual weights as function of node degree (global model, 2500 runs, $n = 1000$, $m = 5$, $\alpha = 0.25$, Borda scoring).

we show average utility and average weight in Fig. 2. From Fig. 2(a) we see that as node degree increases (each node has identical in/out-degree), overall utility tends to increase; moreover most of this increase is due to in large part to an increase in intrinsic utility. Fig. 2(b) also illustrates a strong correlation between degree and agent weight. Nodes with higher degree are more powerful and "influential" in the choice of the consensus alternative. This correlation might be an artifact of the specific social networks we generate. However, the relationship between Figs. 2(a) and (b)—which is independent of the specifics of our experiments—shows that individuals with higher weight tend to prefer the consensus winner more than individuals with lower weight.

The effect of m, k, and scoring rule. We now explore the impact on RSWL of changing the numbers of agents m, the number of initial nodes k when generated the network, and difference between Borda and plurality scoring. We set $\alpha = 0.25$, $n = 1000$, and run 2500 instances for each parameter setting (as above).

Fig. 4 shows average (and maximum, minimum) RSWL for three actual, assumed model combinations for various combinations of rule, m and k, denoted by rule(m, k) (e.g., Plura$(5, 1)$ represents $m = 5$, $k = 1$, and plurality). Comparing Borda$(5, 1)$ and Plura$(5, 1)$, and Borda$(10, 1)$ and Plura$(10, 1)$, we see plurality is more susceptible to social welfare loss than Borda. Increasing m has negligible effect on RSWL when Borda is used, but this is not true of plurality. Surprisingly, increasing k from 1 to 2 decreases RSWL (see Borda$(5, \cdot)$): this occurs because, when $k = 2$, the resulting network is denser since each node has at least two neighbors. This connectivity, causes the number of "very influential" agents to increase; but since weights are normalized (the sum of all weights sums to n), their overall influence decreases as they "share their influence," and weight variance over $\mathcal{N}$ decreases.

Self-loop weight α. When we vary the self-loop weight α, it has a significant effect on RSWL when the actual utility model is global but the intrinsic utility model is assumed. We fix $n = 1000$, $m = 5$, $k = 1$ and vary α over $\{0.05, 0.1, 0.25, 0.5, 0.75\}$ (2500 instances for each setting). Table 4 shows that, for both Borda and plurality, increasing α (i.e., decreasing overall degree of empathy) decreases RSWL.

Number of Iterations of ICE. Finally we examine how the self-loop weight α and Borda/plurality utilities affect the expected number of iterations by the iterated candidate

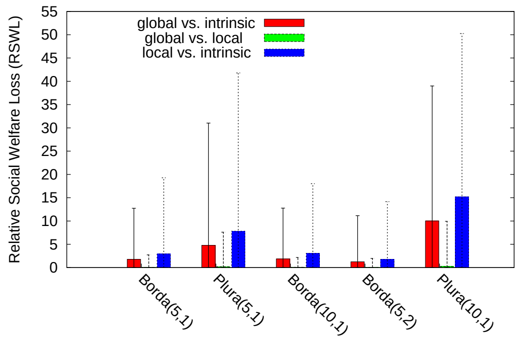


Figure 3: The average (maximum, minimum) RSWL (2500 runs).

	$\alpha = 0.05$	$\alpha = 0.1$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
Borda	15.84%	14.82%	12.73%	9.79%	5.42%	0%
Plurality	39.12%	36.29%	31.02%	22.46%	13.09%	0%

Table 4: Maximum values of RSWL, global vs. intrinsic models.

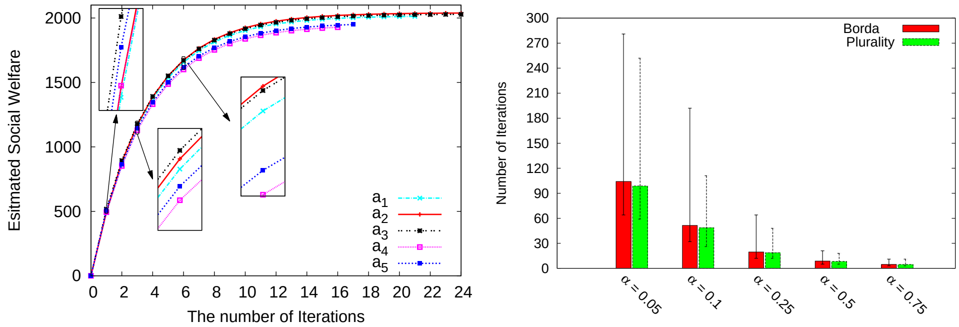
elimination algorithm. We fix $n = 1000$ and $m = 5$, and vary α over $\{0.05, 0.1, 0.25, 0.5, 0.75\}$ (2500 instances). Fig. 4(a) illustrates estimated social welfare for each alternative in one representative instance, with $\alpha = 0.25$ and Borda scoring (this instance of ICE converges in under 2 milliseconds). It converges completely in 24 iterations (n.b. $n = 1000$). Alternatives a_4 and a_5 are eliminated at iterations 16 and 17, respectively; a_1 after 20 iterations; and a_2 after 24 iterations; hence a_3 is optimal. We note that the relative ordering of the alternatives is fixed after 6 iterations (in this instance), which might suggest new methods for early termination.

Fig. 4(b) shows the average (and max, min) number of iterations of ICE for various α, for both Borda and plurality. In all cases, the number of iterations is small compared to the size of the network. ICE is relatively insensitive to the scoring rule, and convergence time decreases dramatically with increasing α, as is typical for iterative algorithms (e.g., for Markov chains). (i.e., for a specific α, the average required iterations is almost the same for Borda and plurality).

5 Concluding Remarks and Future Work

We have presented a new model for social choice situations in which an individual's intrinsic preference for alternatives is combined with their *empathetic* preferences, reflecting their desire to see others satisfied with the selected alternative. Treating a social network as one possible measure of strength of empathetic preference, we developed models and algorithms, for both local and global empathetic settings, that allow one to compute social welfare maximizing outcomes efficiently by weighting the contribution of each agent. Our models have a natural interpretation as empathetic voting models when scoring rules are used. Critically, we require only that individuals specify their intrinsic preferences (and network weights): they need not reason about their neighbor's preferences.

This model, while novel, is merely a starting point for a broader investigation into the role of empathetic preferences in social choice. We are currently exploring more realistic processes for simultaneous generation of networks and preferences that are even better suited to empathetic voting than preferential attachment networks. While our focus has been the choice of a single alternative/winner, our model can also be applied to matching, assignment, and other group decision problems; each will require its own analysis and algorithmic de-



(a) Estimated social welfare over iterations (b) Number of iterations for different α

Figure 4: (a) Estimated social welfare over iterations of ICE for 1 run and (b) average (with maximum and minimum) number of iterations of ICE.

velopments. More importantly is the question of the prevalence and strength of empathetic preferences, the extent to which social network structure is indicative of such preferences, and how one can best discover these preferences in practical settings without an excessive burden on users. Two other important directions are: voting schemes in which agents can specify their tradeoffs between intrinsic and empathetic preference in a more qualitative fashion; and considering the possibility that agents are not truthful and fully aware of their utility functions. These questions require both social scientific and computational insight.

References

[1] S. L. Althaus. Information effects in collective preferences. *Amer. Political Sci. Rev.*, 92(3):545–558, 1998.

[2] A. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, 286:509–512, 1999.

[3] E. Bodine-Baron, C. Lee, A. Chong, B. Hassibi, and A. Wierman. Peer effects and stability in matching markets. *SAGT-10*, pp.117–129, 2011.

[4] P. Boldi, F. Bonchi, C. Castillo, and S. Vigna. Voting in Social Networks. *CIKM'09*, pp.777–786, 2009.

[5] S. Brânzei and K. Larson. Coalitional affinity games and the stability gap. *IJCAI'09*, pp.79–84, 2009.

[6] S. Brânzei and K. Larson. Social distance games. *IJCAI'11*, pp.91–96, 2011.

[7] N. Christakis and J. Fowler. *Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives.* Back Bay Books, 2011.

[8] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* 1990.

[10] D. A. Easley and J. M. Kleinberg. *Networks, Crowds, and Markets.* Cambridge, 2010.

[11] J. H. Fowler and N. A. Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the Framingham Heart Study. *Brit. Med. J.*, 337:a2338, 2008.

[12] N. Haghpanah, N. Immorlica, V. Mirrokni, K. Munagala. Optimal auctions with positive network externalities. *EC'11*, pp.11–20, 2011.

[13] M. O. Jackson. *Social and Economic Networks.* Princeton, Princeton, NJ, 2008.

[14] D. C. Mutz. Cross-cutting social networks: Testing democratic theory in practice. *American Political Science Review*, 96(1):111–126, 2002.

[15] A. E. Sokhey and S. D. McClurg. Social networks and correct voting. Preprint, 2008.

[16] A. Wilson, M. Baccara, A. Imrohoroglu, and L. Yariv. A field study on matching with network externalities. *Behavioral and Quantitative Game Theory (BQGT'10)*, 2010.

Amirali Salehi-Abari
Department of Computer Science
University of Toronto
Email: `abari@cs.toronto.edu`

Craig Boutilier
Department of Computer Science
University of Toronto
Email: `cebly@cs.toronto.edu`

Exploiting Polyhedral Symmetries in Social Choice[1]

Achill Schürmann

Abstract

One way of computing the probability of a specific voting situation under the Impartial Anonymous Culture assumption is via counting integral points in polyhedra. Here, Ehrhart theory can help, but unfortunately the dimension and complexity of the involved polyhedra grows rapidly with the number of candidates. However, if we exploit available polyhedral symmetries, some computations become possible that previously were infeasible. We show this in three well known examples: Condorcet's paradox, Condorcet efficiency of plurality voting and in Plurality voting vs Plurality Runoff.

1 Introduction

Under the Impartial Anonymous Culture (IAC) assumption, the probability for certain election outcomes can be computed by counting integral solutions to a system of linear inequalities, associated to the specific voting event of interest (see for example [GL11]). There exists a rich mathematical theory going back to works of Ehrhart [Ehr67] in the 1960s that helps to deal with such problems. We refer to [BR07] and [Bar08] for an introduction. The connection to Social Choice Theory was discovered by Lepelley et al. [LLS08] and Wilson and Pritchard [WP07]. A few years earlier a similar theory had been described specifically for the social choice context by Huang and Chua [HC00] (see also [Geh02]). Based on Barvinok's algorithm [Bar94] there now exists specialized mathematical software for performing previously cumbersome or practically impossible computations. The first available program was `LattE`, with its newest version `LattE integrale` (see [LDK⁺11a]); alternatives are `barvinok` (see [VB08]) and `Normaliz` (see [BIS12]) which are also usable within the `polymake` framework (see [GJ00]).

The purpose of this note is to shed some light on the possibilities for social choice computations that arise through the use of Ehrhart theory and weighted generalizations of it (see [BBL⁺10]). We in particular show how symmetry of linear systems characterizing certain voting events can be used to reduce computation times, and in some cases, even leads to previously unkown results. As examples, we consider three well studied voting situations with four candidates: *Condorcet's paradox*, the *Condorcet efficiency of plurality voting* and different outcomes in *Plurality vs Plurality Runoff.*

In Section 2 we review some linear models for voting events and introduce some of the used notation. In Section 3 we sketch how counting integral points in polyhedra and Ehrhart's theory can be used to compute probabilities for voting outcomes. In Section 4 we show how the complexity of computations can be reduced by using a symmetry reduced, lower dimensional reformulation. We in particular show how to use integration to obtain exact values for the limiting probability of voting outcomes when the number of voters tends to infinity. As examples, we obtain previously unknown, exact values for two four candidate election events: for the Condorcet efficiency of plurality voting and for Plurality vs Plurality Runoff.

[1]An earlier version of this paper has been published online by *Social Choice and Welfare*, April 13th 2012.

2 Linear systems describing voting situations

Notation

For the start we look at three candidate elections, as everything that will follow can best be motivated and explained in smaller examples. Assume there are n voters, with $n \geq 2$, and each of them has a complete linear (strict) preference order on the three candidates $\mathsf{a}, \mathsf{b}, \mathsf{c}$. We subdivide the voters into six groups

$$(n_{\mathsf{ab}}, n_{\mathsf{ac}}, n_{\mathsf{ba}}, n_{\mathsf{bc}}, n_{\mathsf{ca}}, n_{\mathsf{cb}}), \tag{1}$$

according to their six possible preference orders:

$$\mathsf{abc}(n_{\mathsf{ab}}) \quad \mathsf{acb}(n_{\mathsf{ac}}) \quad \mathsf{bac}(n_{\mathsf{ba}}) \quad \mathsf{bca}(n_{\mathsf{bc}}) \quad \mathsf{cab}(n_{\mathsf{ca}}) \quad \mathsf{cba}(n_{\mathsf{cb}})$$

For example, there are n_{ab} voters that prefer a over b and b over c. We omit the last preference in the index, as it is determined once we know the others. This type of indexing will show to be useful when we reduce the number of variables in Section 4.

The tuple (1) is referred to as a *voting situation*. In an election with

$$n = n_{\mathsf{ab}} + n_{\mathsf{ac}} + n_{\mathsf{ba}} + n_{\mathsf{bc}} + n_{\mathsf{ca}} + n_{\mathsf{cb}} \tag{2}$$

voters, there are $\binom{n+5}{5}$ possible voting situations. We make the simplifying *Impartial Anonymous Culture (IAC) assumption* that each of these voting situations is equally likely to occur.

Condorcet's Paradox

Maybe the most famous voting paradox goes back to the Marquis de Condorcet (1743–1793). He observed that in an election with three or more candidates, it is possible that pairwise comparison of candidates can lead to an intransitive collective choice. For instance, candidate a could be preferred over candidate b, b could be preferred over candidate c and c could be preferred over candidate a. In this case there is no *Condorcet winner*, that is, someone who beats every other candidate by pairwise comparison.

The condition that candidate a is a Condorcet winner can be described via two linear constraints:

$$n_{\mathsf{ab}} + n_{\mathsf{ac}} + n_{\mathsf{ca}} > n_{\mathsf{ba}} + n_{\mathsf{bc}} + n_{\mathsf{cb}} \tag{3} \quad (\ \mathsf{a} \text{ beats } \mathsf{b}\ )$$

$$n_{\mathsf{ab}} + n_{\mathsf{ac}} + n_{\mathsf{ba}} > n_{\mathsf{ca}} + n_{\mathsf{bc}} + n_{\mathsf{cb}} \tag{4} \quad (\ \mathsf{a} \text{ beats } \mathsf{c}\ )$$

The probability of candidate a being a Condorcet winner in an election with n voters can be expressed as the quotient

$$\text{Prob}(n) = \frac{\text{card}\left\{(n_{\mathsf{ab}}, \ldots, n_{\mathsf{cb}}) \in \mathbb{Z}^6_{\geq 0} \text{ satisfying (2), (3), (4) }\right\}}{\binom{n+5}{5}}.$$

The denominator is a polynomial of degree 5 in n. It had been observed by Fishburn and Gehrlein [GF76] (cf. [BB83]) that the numerator shows a similar behavior: Restricting to even or odd n it can be expressed as a degree 5 polynomial in n. The leading coefficient of both polynomials is the same and we approach the same probability for large elections (as n tends to infinity). This *limiting probability* is known to be

$$\lim_{n\to\infty} \text{Prob}(n) = \frac{5}{16}.$$

Having the probability for candidate a being a Condorcet winner, we obtain the probability for a Condorcet paradox (no Condorcet winner exists) as $1 - 3 \cdot \text{Prob}(n)$ with an exact limiting probability of $\frac{1}{16}$.

In a similar way we can determine probabilities for other voting events.

Condorcet efficiency of Plurality voting

If there is a Condorcet winner, there is good reason to consider him to be the voter's choice. However, many common voting rules do not always choose the Condorcet winner even if one exists. This is in particular the case for the widely used plurality voting, where the candidate with a majority of first preferences is elected.

The condition that candidate a is a Condorcet winner but candidate b is the plurality winner can be expressed by the two inequalities (3) and (4), together with the two additional inequalities

$$n_{\mathsf{ba}} + n_{\mathsf{bc}} > n_{\mathsf{ab}} + n_{\mathsf{ac}} \qquad (5) \qquad (\text{ b wins plurality over a })$$
$$n_{\mathsf{ba}} + n_{\mathsf{bc}} > n_{\mathsf{ca}} + n_{\mathsf{cb}} \qquad (6) \qquad (\text{ b wins plurality over c })$$

The *Condorcet efficiency* of a voting rule is the conditional probability that a Condorcet winner is elected if one exists. As there are $3 \cdot 2$ possibilities for choosing a Condorcet winner and another plurality winner, we obtain

$$\text{Prob}(n) = \frac{6 \cdot \text{card}\left\{(n_{\mathsf{ab}}, \ldots, n_{\mathsf{cb}}) \in \mathbb{Z}^6_{\geq 0} \text{ satisfying (2), (3), (4), (5), (6)}\right\}}{3 \cdot \text{card}\left\{(n_{\mathsf{ab}}, \ldots, n_{\mathsf{cb}}) \in \mathbb{Z}^6_{\geq 0} \text{ satisfying (2), (3), (4) }\right\}}$$

for the likelihood of a Condorcet winner being a plurality loser. Again, depending on n being odd or even, one obtains polynomials in n in the denominator and the numerator (see [Geh82]). The exact value of the limit $\lim_{n\to\infty} \text{Prob}(n)$ is $16/135$. Therefore, the Condorcet efficiency of plurality voting with three candidates is $119/135 = 88.\overline{148}\%$.

Plurality vs Plurality Runoff

Plurality Runoff voting is a common practice to overcome some of these "problems" of Plurality voting. It is used in many presidential elections, for example in France. After a first round of plurality voting in which none of the candidates has achieved more than 50% of the votes, the first two candidates compete in a second runoff round.

The condition that candidate b is the plurality winner, but candidate a wins the second round of Plurality Runoff can be expressed by the two inequalities (5) and

$$n_{\mathsf{ab}} + n_{\mathsf{ac}} > n_{\mathsf{ca}} + n_{\mathsf{cb}}, \qquad (7) \qquad (\text{ a wins plurality over c })$$

together with the linear condition (3) that a beats b in a pairwise comparison. The probability that another candidate is chosen in the second round as the number of voters tends to infinity is known to be $71/576 = 12.32638\overline{8}\%$ (see [LLS08]).

Four and more candidates

Having m candidates we can set up similar linear systems in $m!$ variables. For example, in an election with four candidates a, b, c, d we use the 24-dimensional vector $x^t = (n_{\mathsf{abc}}, \ldots, n_{\mathsf{dcb}})$. Here, indices are taken in lexicographical order. The condition that a is a Condorcet winner is described by the three inequalities that imply a beats b, a beats c and a beats d in a pairwise comparison. As linear systems with 24 variables become hard to grasp, it is convenient to use matrices for their description. We are interested in all non-negative integral (column) vectors x satisfying the matrix inequality $Ax > 0$ for the matrix $A \in \mathbb{Z}^{3\times 24}$ with entries

```
 1  1  1  1  1  1 -1 -1 -1 -1 -1 -1  1  1 -1 -1  1 -1  1  1 -1 -1  1 -1
 1  1  1  1  1  1  1  1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1  1  1  1 -1 -1 -1
 1  1  1  1  1  1  1  1  1 -1 -1 -1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

(8)

3 Likelihood of voting situations and Ehrhart's theory

Integral points in polyhedral cones

In order to deal with an arbitrary number of candidates, let us put the example above in a slightly more general context. In any of the three voting examples, the voting situations of interest lie in a *polyhedral cone*, that is, in a set $\mathcal{P}$ of points in $\mathbb{R}^d$ (with $d = 6$ or $d = 24$ in case of three or four candidate elections) satisfying a finite number of homogeneous linear inequalities. In addition to the strict inequalities which are different in each of the examples, the condition that the variables n_i are non-negative can be expressed by the homogeneous linear inequalities $n_i \geq 0$.

Let $\mathcal{P}, \mathcal{S} \subset \mathbb{R}^d$ denote two d-dimensional *polyhedral cones*, each defined by some homogeneous linear (possibly strict) inequalities. We may assume that $\mathcal{P}$ is contained in $\mathcal{S}$ and that both polyhedral cones are contained in the orthant $\mathbb{R}^d_{\geq 0}$. If we are interested in elections with n voters, we consider the voting situations (integral vectors) in the intersection of $\mathcal{P}$ and $\mathcal{S}$ with the affine subspace

$$L_n^d = \left\{ (n_1, \ldots, n_d) \in \mathbb{R}^d \mid \sum_{i=1}^{d} n_i = n \right\}.$$

The *expected frequency* of voting situations being in $\mathcal{P}$ among voting situations in $\mathcal{S}$ is then expressed by

$$\operatorname{Prob}(n) = \frac{\operatorname{card}\left(\mathcal{P} \cap L_n^d \cap \mathbb{Z}^d\right)}{\operatorname{card}\left(\mathcal{S} \cap L_n^d \cap \mathbb{Z}^d\right)}. \tag{9}$$

When estimating the probability of candidate a being a Condorcet winner for instance, the homogeneous polyhedral cone $\mathcal{S}$ is simply the non-negative orthant $\mathbb{R}^d_{\geq 0}$ described by the linear inequalities $n_i \geq 0$. In that case the denominator is known to be equal to

$$\binom{n+d-1}{d-1}.$$

This is a polynomial in n of degree $d-1$ (the dimension of $L_n^d \cap \mathcal{S}$).

Ehrhart theory

By Ehrhart's theory, the number of integral solutions in a polyhedral cone intersected with L_n^d can be expressed by a *quasi-polynomial* in n. Roughly speaking, a quasi-polynomial is simply a finite collection $p_1(n), \ldots, p_k(n)$ of polynomials, such that the number of voting situations is given by $p_i(n)$ if $i \equiv n \mod k$.

The degree of the polynomial is equal to the dimension of the polyhedral cone intersected with L_n^d. In the voting events considered here this dimension is always equal to $d-1$. So in the examples with three candidates their degree is always 5. The number k of different polynomials depends on the linear inequalities involved. For the Condorcet paradox we have $k = 2$ polynomials $p_1(n)$ and $p_2(n)$, where $p_1(n)$ gives the answer for odd n ($1 \equiv n \mod 2$) and $p_2(n)$ gives the answer for even n ($0 \equiv 2 \equiv n \mod 2$). For Condorcet efficiency we have $k = 6$ (see [Geh02]) and for Plurality vs Plurality Runoff we have $k = 12$ (see [LLS08]).

Given a polyhedral cone $\mathcal{P}$, the quasi-polynomial $q(n) = \operatorname{card}\left(\mathcal{P} \cap L_n^d \cap \mathbb{Z}^d\right)$ can be explicitly computed using software packages like `LattE integrale` [`latte`] or `barvinok` [`barvinok`]. The result for the polyhedral cone $\mathcal{P}$ describing candidate a as the Condorcet winner could look like

```
  1/384 * n^5
+ ( 1/64 * { 1/2 * n } + 1/32 ) * n^4
+ ( 17/96 * { 1/2 * n } + 13/96 ) * n^3
+ ( 23/32 * { 1/2 * n } + 1/4 ) * n^2
+ ( 233/192 * { 1/2 * n } + 1/6 ) * n
+ ( 45/64 * { 1/2 * n } + 0 )
```

The curly brackets $\{\cdots\}$ mean the fractional part of the enclosed number, allowing to write the quasi-polynomial in a closed form. In this example we get different polynomials for odd and even n. Note that the leading coefficient (the coefficient of n^5) is in both cases the same. By Ehrhart's theory this is always the case, as it is equal to the *relative volume* of the polyhedron $\mathcal{P} \cap L_1^d$. That is, it is equal to a $\sqrt{d}$-multiple of the standard Lebesgue measure on the affine space L_1^d. The measure is normalized so that the space contains one integral point per unit volume.

One technical obstacle using software like `LattE integrale` or `barvinok` is the use of polyhedral cones described by a mixture of strict and non-strict inequalities. As the software assumes the input to have only non-strict inequalities or equality conditions, one has to avoid the use of strict inequalities. A simple way to achieve this is the replacement of strict inequalities $x > 0$ by non-strict ones $x \geq 1$, in case x is known to be integral. For instance, if x is a linear expression with integer coefficients, and if we are interested in integral solutions as in our examples, this is a possible reformulation.

Altogether, by obtaining quasi-polynomials for numerator and denominator in (9) we get an explicit formula for $\mathrm{Prob}(n)$ via Erhart's theory.

Limiting probabilities via integration

If we want to compute the exact value of $\lim_{n\to\infty} \mathrm{Prob}(n)$ as n tends to infinity, we can use volume computations without using Ehrhart's theory. As mentioned above, the leading coefficients of denominator and numerator correspond to the relative volumes of the sets $\mathcal{P} \cap L_1$ and $\mathcal{S} \cap L_1$:

$$\lim_{n\to\infty} \mathrm{Prob}(n) \;=\; \lim_{n\to\infty} \frac{\mathrm{card}\left(\mathcal{P} \cap L_1^d \cap (\mathbb{Z}/n)^d\right)}{\mathrm{card}\left(\mathcal{S} \cap L_1^d \cap (\mathbb{Z}/n)^d\right)} \;=\; \frac{\mathrm{relvol}\left(\mathcal{P} \cap L_1^d\right)}{\mathrm{relvol}\left(\mathcal{S} \cap L_1^d\right)}$$

In fact, as long as we use the same measure to evaluate the numerator and the denominator, it does not matter what multiple of the standard Lebesgue measure we use to compute volume on the affine space L_1^d. The exact relative volume can be computed using `LattE integrale`. Alternatives are for example `Normaliz` (see [`normaliz`]) or `vinci` (see [BEF00]). Exact computations can be quite involved in higher dimensions (cf. [DF88]). In such cases it is sometimes only possible to compute an approximation, using *Monte Carlo methods* for instance.

4 Reducing the dimension by exploiting polyhedral symmetries

In many models the involved linear systems and polyhedra are quite symmetric. In particular, permutations of variables may lead to equivalent linear systems describing the same polyhedron. Such symmetries are often visible in smaller examples and can automatically be determined for larger problems, for instance by our software `SymPol` (see [RS10]). In the three examples described in Section 2, we can exploit such symmetries to reduce the complexity of computations.

Condorcet's paradox

In case of a being a Condorcet winner in a three candidate election, the variables n_{ab} and n_{ac} occur pairwise (as $n_{\mathsf{ac}} + n_{\mathsf{ab}}$) in inequalities (3), (4) and in equation (2). The same is true for n_{bc} and n_{cb}. By introducing new variables $n_{\mathsf{a}} = n_{\mathsf{ac}} + n_{\mathsf{ab}}$ and $n_{*\mathsf{a}} = n_{\mathsf{bc}} + n_{\mathsf{cb}}$ we can reduce the dimension of the linear system to only four variables:

$$\begin{aligned} n_{\mathsf{a}} + n_{\mathsf{ca}} - n_{*\mathsf{a}} - n_{\mathsf{ba}} &> 0 \\ n_{\mathsf{a}} + n_{\mathsf{ba}} - n_{*\mathsf{a}} - n_{\mathsf{ca}} &> 0 \\ n_{\mathsf{a}} + n_{\mathsf{ca}} + n_{*\mathsf{a}} + n_{\mathsf{ba}} &= n \\ n_{\mathsf{a}}, n_{*\mathsf{a}}, n_{\mathsf{ba}}, n_{\mathsf{ca}} &\geq 0. \end{aligned}$$

The index a indicates that we group all variables which carry candidate a as their first preference and index $*\mathsf{a}$ stands for grouping of all variables with candidate a ranked last. In the reduced linear system each 4-tuple $(n_{\mathsf{a}}, n_{*\mathsf{a}}, n_{\mathsf{ba}}, n_{\mathsf{ca}})$ represents several voting situations, previously described by 6-tuples. For n_{a} we have $(n_{\mathsf{a}} + 1)$ different possibilities of non-negative integral tuples $(n_{\mathsf{ac}}, n_{\mathsf{ab}})$. Similar is true for $n_{*\mathsf{a}}$. Together we have

$$(n_{\mathsf{a}} + 1)(n_{*\mathsf{a}} + 1)$$

voting situations with three candidates represented by each non-negative integral vector $(n_{\mathsf{a}}, n_{*\mathsf{a}}, n_{\mathsf{ba}}, n_{\mathsf{ca}})$.

In the four candidate case it is possible to obtain a similar reformulation by grouping among 24 variables. We introduce a new variable for sets of variables having same coefficients in the linear system. Having a matrix representation as in (8), this kind of special symmetry in the linear system is easy to find by identifying equal columns. Introducing a new variable for each set of equal columns we get

$$\begin{aligned} n_{\mathsf{a}} - n_{\mathsf{ba}} + n_{\mathsf{ca}} + n_{\mathsf{da}} + n_{*\mathsf{ab}} - n_{*\mathsf{ac}} - n_{*\mathsf{ad}} - n_{*\mathsf{a}} &> 0 \\ n_{\mathsf{a}} + n_{\mathsf{ba}} - n_{\mathsf{ca}} + n_{\mathsf{da}} - n_{*\mathsf{ab}} + n_{*\mathsf{ac}} - n_{*\mathsf{ad}} - n_{*\mathsf{a}} &> 0 \\ n_{\mathsf{a}} + n_{\mathsf{ba}} + n_{\mathsf{ca}} - n_{\mathsf{da}} - n_{*\mathsf{ab}} - n_{*\mathsf{ac}} + n_{*\mathsf{ad}} - n_{*\mathsf{a}} &> 0 \end{aligned} \tag{10}$$

These three inequalities describe voting situations in which candidate a beats candidates b, c and d each in a pairwise comparison. As in all of our examples, we additionally have the condition that the involved variables add up to n and that all of them are non-negative.

As before, the used indices of variables reflect which voter preferences are grouped. As in the three candidate case, n_{a} and $n_{*\mathsf{a}}$ denote the number of voters with candidate a being their first and last preference respectively. Similarly, xy and $*\mathsf{yx}$ in the index indicate that voters with preference order starting with x, y and ending with y, x have been combined.

Using our software `SymPol` [`sympol`] one easily checks that the original system with 24 variables has a symmetry group of order 199065600. The new reduced system with 8 variables, obtained through the above grouping of variables, turns out to have a symmetry group of order 6 only. So most of the symmetry in the original system is of the simple form that is detectable through equal columns in a matrix representation. The remaining 6-fold symmetry comes from the possibility to arbitrarily permute the variables $n_{\mathsf{ba}}, n_{\mathsf{ca}}, n_{\mathsf{da}}$ when at the same time the variables $n_{*\mathsf{ab}}, n_{*\mathsf{ac}}, n_{*\mathsf{ad}}$ are permuted accordingly. This symmetry is due to the fact that candidates b, c and d are equally treated in the linear system (10). The two new variables n_{a} and $n_{*\mathsf{a}}$ each combine six of the former variables. The other six new variables each combine two former ones.

Weighted counting

In general, if we group more than two variables, say if we substitute the sum of k variables $n_1 + \ldots + n_k$ by a new variable N, we have to include a factor of

$$\binom{N+k-1}{k-1}$$

when counting voting situations via N. If we substitute d variables $(n_1, \ldots, n_d)$ by D new variables $(N_1, \ldots, N_D)$, say by setting N_i to be the sum of k_i of the variables n_j, for $i = 1, \ldots, D$, then we count for each D-tuple

$$p(N_1, \ldots, N_D) = \prod_{i=1}^{D} \binom{N_i + k_i - 1}{k_i - 1} \tag{11}$$

many voting situations.

In the example above, with four candidates and candidate a being the Condorcet winner, we have $d = 24$, $D = 8$ and we obtain a degree 16 polynomial

$$\binom{n_{\mathsf{a}}+5}{5}(n_{\mathsf{ba}}+1)(n_{\mathsf{ca}}+1)(n_{\mathsf{da}}+1)(n_{*\mathsf{ab}}+1)(n_{*\mathsf{ac}}+1)(n_{*\mathsf{ad}}+1)\binom{n_{*\mathsf{a}}+5}{5}$$

to count voting situations for each 8-tuple

$$(n_{\mathsf{a}}, n_{\mathsf{ba}}, n_{\mathsf{ca}}, n_{\mathsf{da}}, n_{*\mathsf{ab}}, n_{*\mathsf{ac}}, n_{*\mathsf{ad}}, n_{*\mathsf{a}}) \,.$$

Geometrically, the polyhedral cone $\mathcal{P} \subset \mathbb{R}^d$ is replaced by a new polyhedral cone $\mathcal{P}' \subset \mathbb{R}^D$ in a lower dimension. As the counting is changed we obtain for the probability (9) of voting situations in $\mathcal{P}$ among those in $\mathcal{S}$:

$$\mathrm{Prob}(n) \;=\; \frac{\displaystyle\sum_{x \in \mathcal{P} \cap L_n^d \cap \mathbb{Z}^d} 1}{\displaystyle\sum_{x \in \mathcal{S} \cap L_n^d \cap \mathbb{Z}^d} 1} \;=\; \frac{\displaystyle\sum_{y \in \mathcal{P}' \cap L_n^D \cap \mathbb{Z}^D} p(y)}{\displaystyle\sum_{y \in \mathcal{S}' \cap L_n^D \cap \mathbb{Z}^D} p(y)}. \tag{12}$$

Here, $\mathcal{S}'$ is equal to the corresponding homogeneous polyhedral cone obtained from $\mathcal{S} \subset \mathbb{R}^d$, and $p(y)$ is the polynomial (11) in D variables. In the example of Condorcet's paradox, $\mathcal{S}'$ is simply equal to the full orthant $\mathbb{R}^D_{\geq 0}$.

As seen in Section 3, we can use Ehrhart's theory to determine an explicit formula for $\mathrm{Prob}(n)$. The right hand side of the formula above suggests that we can do this also via *weighted lattice point counting* in dimension D. A corresponding Ehrhart-type theory has recently been considered (see [BBL+10]). A first implementation is available in the package `barvinok` via the command `barvinok_summate`. We successfully tested the software on some reformulations of three candidate elections, but so far `barvinok` seems not capable to do computations for the four candidate case. However, there still seems quite some improvement possible in the current implementation (personal communication with Sven Verdoolaege). It can be expected that future versions of `LattE integrale` will be capable of these computations (personal communication with Matthias Köppe). It appears to be "just" a matter of implementing the ideas in [BBL+10].

We note that, theoretically, it can generally be expected that weighted counting over a smaller dimensional polyhedron is faster than unweighted counting over a corresponding high dimensional polyhedron. However, due to fact that a suitable implementation for weighted counting is not available at the moment, latter approach may practically still be a good choice. For instance, the latest version of `Normaliz` (July 2012) appears to be capable to obtain the Ehrhart quasi-polynomials for the 23-dimensional polyhedra considered in this note (personal communication with Winfried Bruns and Bogdan Ichim).

Limiting probabilities via integration

If we want to compute the exact value of $\lim_{n\to\infty} \mathrm{Prob}(n)$ we may use integration. Using (12) we get through substitution of $y = nz$:

$$\begin{aligned}
\lim_{n\to\infty} \mathrm{Prob}(n) &= \lim_{n\to\infty} \frac{\sum\limits_{y\in\mathcal{P}'\cap L_n^D\cap\mathbb{Z}^D} p(y)}{\sum\limits_{y\in\mathcal{S}'\cap L_n^D\cap\mathbb{Z}^D} p(y)} = \lim_{n\to\infty} \frac{\sum\limits_{z\in\mathcal{P}'\cap L_1^D\cap(\mathbb{Z}/n)^D} p(nz)}{\sum\limits_{z\in\mathcal{S}'\cap L_1^D\cap(\mathbb{Z}/n)^D} p(nz)} \\
&= \lim_{n\to\infty} \frac{\sum\limits_{z\in\mathcal{P}'\cap L_1^D\cap(\mathbb{Z}/n)^D} p(nz)/n^{\deg p}}{\sum\limits_{z\in\mathcal{S}'\cap L_1^D\cap(\mathbb{Z}/n)^D} p(nz)/n^{\deg p}} = \frac{\int_{\mathcal{P}'\cap L_1^D} \mathrm{lt}(z)\, dz}{\int_{\mathcal{S}'\cap L_1^D} \mathrm{lt}(z)\, dz}.
\end{aligned}$$

Here, the division of numerator and denominator by a degree of p ($\deg p$) power of n shows that the integrals on the right are taken over the leading term $\mathrm{lt}(z)$ of the polynomial $p(z)$ only. Thus determining the exact limiting probability is achieved by integrating a degree $d - D$ monomial over a bounded polyhedron (*polytope*) in the $(D-1)$-dimensional affine space L_1^D. We refer to [LDK+11b] for background on efficient integration methods (cf. [BBL+11] and [Sch98]).

As in the case of relative volume computations in dimension d, the integral is taken with respect to the relative Lebesgue measure – here on the affine space L_1^D. In fact, as we are computing a quotient, any measure being a multiple of the standard Lebesgue measure on L_1^D will give the same value.

For the example with candidate a being a Condorcet winner in a four candidate election, the leading term to be integrated is

$$n_{\mathsf{a}}^5 \cdot n_{\mathsf{ba}} \cdot n_{\mathsf{ca}} \cdot n_{\mathsf{da}} \cdot n_{*\mathsf{ab}} \cdot n_{*\mathsf{ac}} \cdot n_{*\mathsf{ad}} \cdot n_{*\mathsf{a}}^5,$$

which is much simpler than the full polynomial. Integrating this polynomial over the reduced 8-dimensional polyhedron can be done using `LattE integrale` (called with option `valuation=integrate`). In this way one obtains in a few seconds an exact value of 1717/2048 for the probability that a Condorcet winner exists (as n tends to infinity). This value corresponds to the one obtained by Gehrlein in [Geh01] and serves as a test case for our method. The corresponding volume computation with `LattE integrale` (called with option `valuation=volume`) in 24 variables did not finish after several weeks of computation. This is due to the fact that triangulating a 24-dimensional polyhedron is much more involved than integration over a corresponding lower dimensional polyhedron (of dimension 8 in this case). However, Winfried Bruns, Bogdan Ichim and Christof Söger report (May 2012) that the 24-dimensional volume computation is doable with the newest version of their software `Normaliz` (see [`normaliz`]). Nevertheless, their volume computation, using sophisticated heuristics for triangulations (see [BIS12]), is still much slower than the corresponding integration over the 8-dimensional polyhedron.

In a similar way we can deal with other voting situations as well.

Condorcet efficiency of plurality voting

Assuming candidate a is a Condorcet winner, but candidate b wins a plurality voting, we obtain a reduced system in the three candidate case with five variables:

$$\begin{array}{rcl} n_{\mathsf{a}} - n_{\mathsf{ba}} - n_{\mathsf{bc}} - n_{\mathsf{cb}} + n_{\mathsf{ca}} & > & 0 \\ n_{\mathsf{a}} + n_{\mathsf{ba}} - n_{\mathsf{bc}} - n_{\mathsf{cb}} - n_{\mathsf{ca}} & > & 0 \\ -n_{\mathsf{a}} + n_{\mathsf{ba}} + n_{\mathsf{bc}} & > & 0 \\ n_{\mathsf{ba}} + n_{\mathsf{bc}} - n_{\mathsf{cb}} - n_{\mathsf{ca}} & > & 0 \end{array}$$

Here the only reduction is the grouping $n_{\mathsf{a}} = n_{\mathsf{ab}} + n_{\mathsf{ac}}$. The corresponding polynomial weight is $n_{\mathsf{a}} + 1$.

The four candidate case is more involved. The linear system with 24 variables has a comparatively small symmetry group of order 92160. We can group six variables into n_{a}. Taking the reduced system (10) of three inequalities with 8 variables (modeling that candidate a is a Condorcet winner) we have to add three inequalities for the condition that candidate b wins plurality. These can be shortly described by $n_{\mathsf{b}} > n_{\mathsf{a}}, n_{\mathsf{c}}, n_{\mathsf{d}}$, but a grouping of variables in n_{b}, n_{c} and n_{d} is incompatible with the other three conditions. Instead we use new variables $n_{\mathsf{b*a}}$, $n_{\mathsf{c*a}}$ and $n_{\mathsf{d*a}}$ (in (10) combined in $n_{*\mathsf{a}}$) for preferences in which a is ranked last. Additionally we have to keep the variables where candidate a is ranked third (in (10) combined in $n_{*\mathsf{ab}}$, $n_{*\mathsf{ac}}$, $n_{*\mathsf{ad}}$).

In the three inequalities (10) we can simply substitute $n_{*\mathsf{a}}$ by $n_{\mathsf{b*a}} + n_{\mathsf{c*a}} + n_{\mathsf{d*a}}$ and $n_{*\mathsf{ad}}$, $n_{*\mathsf{ac}}$ and $n_{*\mathsf{ab}}$ by $n_{\mathsf{bca}} + n_{\mathsf{cba}}$, $n_{\mathsf{bda}} + n_{\mathsf{dba}}$ and $n_{\mathsf{cda}} + n_{\mathsf{dca}}$. The additional three linear inequalities for candidate b being a plurality winner are then:

$$\begin{array}{llcl} n_{\mathsf{b*a}} + n_{\mathsf{ba}} + n_{\mathsf{bca}} + n_{\mathsf{bda}} & - n_{\mathsf{a}} & > & 0 \\ n_{\mathsf{b*a}} + n_{\mathsf{ba}} + n_{\mathsf{bca}} + n_{\mathsf{bda}} & - n_{\mathsf{c*a}} - n_{\mathsf{ca}} - n_{\mathsf{cba}} - n_{\mathsf{cda}} & > & 0 \\ n_{\mathsf{b*a}} + n_{\mathsf{ba}} + n_{\mathsf{bca}} + n_{\mathsf{bda}} & - n_{\mathsf{d*a}} - n_{\mathsf{da}} - n_{\mathsf{dba}} - n_{\mathsf{dca}} & > & 0 \end{array}$$

This reduced linear system has 6 inequalities for 13 variables. It still has a symmetry of order 2 coming from an interchangeable role of candidates c and d. The degree 11 polynomial used for integration is

$$n_{\mathsf{a}}^5 \cdot n_{\mathsf{ba}} \cdot n_{\mathsf{ca}} \cdot n_{\mathsf{da}} \cdot n_{\mathsf{b*a}} \cdot n_{\mathsf{c*a}} \cdot n_{\mathsf{d*a}}.$$

With it, using `LattE integrale`, we obtain an exact limit of

$$\frac{10658098255011916449318509}{14352135440302080000000000} = 74.261410\ldots\%$$

for the Condorcet efficiency of plurality voting with four candidates. To the best of our knowledge this value has not been computed before.

Plurality vs Plurality Runoff

The case of Plurality vs Plurality Runoff has a high degree of symmetry. For three candidates we obtain a reduced four dimensional reformulation:

$$\begin{array}{rcl} n_{\mathsf{b}} - n_{\mathsf{a}} & > & 0 \\ n_{\mathsf{a}} - n_{\mathsf{ca}} - n_{\mathsf{cb}} & > & 0 \\ n_{\mathsf{a}} + n_{\mathsf{ca}} - n_{\mathsf{b}} - n_{\mathsf{cb}} & > & 0 \end{array}$$

Counting is done via the polynomial weight $(n_{\mathsf{a}} + 1)(n_{\mathsf{b}} + 1)$. Integration of $n_{\mathsf{a}} n_{\mathsf{b}}$ over the corresponding 3-dimensional polyhedron yields the known limiting probability.

If we consider elections with m candidates, $m \geq 4$, we can set up a linear system with only $2(m-1)$ variables and m inequalities. We denote the candidates by a, b and $\mathsf{c_i}$ for $i = 1, \ldots, m-2$:

$$
\begin{array}{rrcl}
 & n_{\mathsf{b}} - n_{\mathsf{a}} & > & 0 \\
\text{For } i = 1, \ldots, m-2: & n_{\mathsf{a}} - n_{\mathsf{c_i}\cdot\mathsf{a}\cdot\mathsf{b}} - n_{\mathsf{c_i}\cdot\mathsf{b}\cdot\mathsf{a}} & > & 0 \\
 & n_{\mathsf{a}} + \sum_{i=1}^{m-2} n_{\mathsf{c_i}\cdot\mathsf{a}\cdot\mathsf{b}} - n_{\mathsf{b}} - \sum_{i=1}^{m-2} n_{\mathsf{c_i}\cdot\mathsf{b}\cdot\mathsf{a}} & > & 0
\end{array}
$$

The first two lines model that candidate b wins plurality over candidate a and that candidate a is second, winning over candidates $\mathsf{c_i}$, for $i = 1, \ldots, m-2$. The last inequality models the condition that candidate a beats b in a pairwise comparison. The variable $n_{\mathsf{c_i}\cdot\mathsf{a}\cdot\mathsf{b}}$ gives the number of voters with candidate $\mathsf{c_i}$ being their first preference and candidate a being ranked before candidate b. Similarly, $n_{\mathsf{c_i}\cdot\mathsf{b}\cdot\mathsf{a}}$ is the number of voters with first preference $\mathsf{c_i}$ and candidate b being ranked before candidate a. We use "$\cdot$" to denote any ordering of candidates; in contrast to "$*$" used before we also allow an empty list here. For both variables, $n_{\mathsf{c_i}\cdot\mathsf{a}\cdot\mathsf{b}}$ and $n_{\mathsf{c_i}\cdot\mathsf{b}\cdot\mathsf{a}}$, we group $(m-1)!/2$ of the $m!$ former variables. The new variables n_{a} and n_{b} both represent $(m-1)!$ former variables. Therefore, counting is adapted using the polynomial weight

$$
(n_{\mathsf{a}} \cdot n_{\mathsf{b}})^{(m-1)!-1} \cdot \prod_{i=1}^{m-2} (n_{\mathsf{c_i}\cdot\mathsf{a}\cdot\mathsf{b}} \cdot n_{\mathsf{c_i}\cdot\mathsf{b}\cdot\mathsf{a}})^{(m-1)!/2-1}
$$

of degree $m! - 2m + 2$.

The above inequalities assume that candidates b and a are ranked first and second in a plurality voting. So having the probability for the corresponding voting situations, we have to multiply by $m(m-1)$ to get the overall probability of a plurality winner losing in a second Plurality Runoff round.

For four candidates ($m = 4$) we obtain an exact limiting probability of

$$
\frac{2988379676768359}{12173449145352192} = 24.548339\ldots\%.
$$

This result can be obtained using the weighted, dimension-reduced problem with `LattE integrale`, as well as by a relative volume computation in 24 variables. However, the latter is a few hundred times slower than integration over the dimension reduced polyhedron. A similar result from a volume computation is obtained in [LDK+11b].

To be certain about our new results, we computed the value above, as well as the likelihood for the existence of a Condorcet winner, with a fully independent `Maple` calculation, using the package `Convex` (see [`convex`]). For it, we first obtained a *triangulation* (non-overlapping union of *simplices*) of the dimension-reduced polyhedron and then applied symbolic integration to each simplex.

We also tried to solve the five candidate case, where the polyhedron is only 7-dimensional (in 8 variables). The integration of a polynomial of degree 112, however, seems a bit too difficult for the currently available technology. Nevertheless it seems that we are close to obtain exact five candidate results as well.

5 Conclusions

Using symmetry of linear systems we can obtain symmetry reduced lower dimensional reformulations. These allow to compute exact limiting probabilities for large elections with four candidates. In this work we only gave a few starting examples. Similar calculations are possible for many other voting situations as well. Even during the work on this project, the software packages `LattE integrale` and `Normaliz` for corresponding polyhedral computations have introduced substantial improvements. We can look forward to capabilities of future versions.

At the moment, for elections with five or more candidates further ideas seem necessary. One possibility to reduce the complexity of computations further is the use of additional symmetries which remain in our reduced systems.

References

[Bar94] A. Barvinok, *A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed*, Math. Oper. Res. **19** (1994), no. 4, 769–779.

[Bar08] ———, *Integer points in polyhedra*, Zurich Lectures in Advanced Mathematics, European Mathematical Society (EMS), Zürich, 2008.

[BBL+10] V. Baldoni, N. Berline, J.A. De Loera, M. Köppe, and M. Vergne, *Computation of the highest coefficients of weighted Ehrhart quasi-polynomials of rational polyhedra*, Found. Comput. Math., to appear, preprint at http://arxiv.org/abs/1011.1602v1.

[BBL+11] ———, *How to integrate a polynomial over a simplex*, Math. Comp. **80** (2011), no. 273, 297–325.

[BB83] S. Berg and B. Bjurulf, *A note on the paradox of voting: anonymous preference profiles and May's formula*, Public Choice **40** (1983), 307–316.

[BIS12] W. Bruns, B. Ichim and C. Sger, *The power of pyramid decomposition in Normaliz*, preprint at `http://arxiv.org/abs/1206.1916`.

[BEF00] B. Büeler, A. Enge, and K. Fukuda, *Exact volume computation for polytopes: a practical study*, Polytopes—combinatorics and computation (Oberwolfach, 1997), DMV Sem., vol. 29, Birkhäuser, Basel, 2000, pp. 131–154.

[BR07] M. Beck and S. Robins, *Computing the continuous discretely*, Undergraduate Texts in Mathematics, Springer, New York, 2007, Integer-point enumeration in polyhedra.

[LDK+11a] J.A. De Loera, B. Dutra, M. Köppe, S. Moreinis, G. Pinto, and J. Wu, *A users guide for* `LattE integrale` *v1.5*, 2011, available at `http://www.math.ucdavis.edu/~latte/`.

[LDK+11b] ———, *Software for Exact Integration of Polynomials over Polyhedra*, 2011, preprint at http://arxiv.org/abs/1108.0117v2.

[DF88] M.E. Dyer and A.M. Frieze, *On the complexity of computing the volume of a polyhedron*, SIAM J. Comput. **17** (1988), no. 5, 967–974.

[Ehr67] E. Ehrhart, *Sur un problème de géométrie diophantienne linéaire. I. Polyèdres et réseaux*, J. Reine Angew. Math. **226** (1967), 1–29.

[Geh82] W.V. Gehrlein, *Condorcet efficiency of constant scoring rules*, Mathematical Social Sciences **2** (1982), no. 2, 123–130.

[Geh01] ———, *Condorcet winners on four candidates with anonymous voters*, Economics Letters **71** (2001), 335–340.

[Geh02] ———, *Obtaining representations for probabilities of voting outcomes with effectively unlimited precision integer arithmetic*, Soc. Choice Welf. **19** (2002), no. 3, 503–512.

[GF76] W.V. Gehrlein and P.C. Fishburn, *The probability of the paradox of voting: a computable solution*, J. Econom. Theory **13** (1976), no. 1, 14–25.

[GJ00] E. Gawrilow and M. Joswig, *polymake: a framework for analyzing convex polytopes*, Polytopes — Combinatorics and Computation (Gil Kalai and Günter M. Ziegler, eds.), Birkhäuser, 2000, pp. 43–74.

[GL11] W.V. Gehrlein and D. Lepelley, *Voting paradoxes and group coherence*, Studies in Choice and Welfare, Springer, Heidelberg, 2011, The Condorcet efficiency of voting rules.

[HC00] H.C. Huang and Vincent C.H. Chua, *Analytical representation of probabilities under the IAC condition*, Soc. Choice Welf. **17** (2000), no. 1, 143–155.

[LLS08] D. Lepelley, A. Louichi, and H. Smaoui, *On Ehrhart polynomials and probability calculations in voting theory*, Soc. Choice Welf. **30** (2008), no. 3, 363–383.

[RS10] T. Rehn and A. Schürmann, *C++ tools for exploiting polyhedral symmetries*, Mathematical Software ICMS 2010, Lecture Notes in Comput. Sci., vol. 6327, Springer, Berlin, 2010, pp. 295–298.

[Sch98] M. Schechter, *Integration over a polyhedron: an application of the Fourier-Motzkin elimination method*, Amer. Math. Monthly **105** (1998), no. 3, 246–251.

[VB08] S. Verdoolaege and M. Bruynooghe, *Algorithms for weighted counting over parametric polytopes: A survey and a practical comparison*, 2008, Eighth ACES Symposium, Edegem, Belgium.

[WP07] M.C. Wilson and G. Pritchard, *Probability calculations under the IAC hypothesis*, Math. Social Sci. **54** (2007), no. 3, 244–256.

Software

[barvinok] `barvinok` *by S. Verdoolaege*, ver. 0.34 (2011), `http://freshmeat.net/projects/barvinok`

[convex] `Convex` *by M. Franz*, ver. 1.1.3 (2009), `http://www.math.uwo.ca/~mfranz/convex/`

[latte] `LattE integrale` *by J.A. DeLoera, M. Köppe et al.*, ver. 1.5 (2011), `http://www.math.ucdavis.edu/~latte/`

[normaliz] `Normaliz` *by W. Bruns, B. Ichim, and C. Söger*, ver. 2.8 (July 2012), `http://www.mathematik.uni-osnabrueck.de/normaliz/`

[sympol] `SymPol` *by T. Rehn and A. Schürmann*, ver. 0.1.7 (April 2012), `http://www.geometrie.uni-rostock.de/software/`

Achill Schürmann, Institute of Mathematics, University of Rostock, 18051 Rostock, Germany
Email: `achill.schuermann@uni-rostock.de`

On Elections with Robust Winners

Dmitry Shiryaev, Lan Yu, Edith Elkind

Abstract

We study the sensitivity of election outcomes to small changes in voters' preferences. We assume that a voter may err by swapping two adjacent candidates in his vote; we would like to check whether the election outcome would remain the same given up to δ errors. We describe polynomial-time algorithms for this problem for all scoring rules as well as for the Condorcet rule. We are also interested in identifying elections that are maximally robust with respect to a given voting rule. We define the robustness radius of an election with respect to a given voting rule as the maximal number of errors that can be made without changing the election outcome; the robustness of a voting rule is defined as the robustness radius of the election that is maximally robust with respect to this rule. We derive bounds on the robustness of various voting rules, including Plurality, Borda, and Condorcet.

1 Introduction

Voting provides a convenient method for preference aggregation in heterogeneous groups of agents: the group members report how they order the available alternatives (from the most preferred one to the least preferred one), and a voting rule is used to select a winner. There is a wide variety of voting rules that can be used for this purpose, with each of these rules encoding a certain approach to aggregating the preferences of the group members. Clearly, for a voting rule to work as intended, it has to be the case that every voter can reliably submit a ranking that fully reflects his opinion of the available alternatives. However, it is not realistic to assume that this is always the case.

Indeed, there are two main reasons for submitting an erroneous vote. First, the voters may be unable to invest sufficient time and resources in investigating the properties of all the available alternatives, and, as a result, they may err by ordering fairly similar alternatives in a way that deviates from the one they would have chosen if they were to study their options in more detail. Second, voters can make mistakes when filling out their ballots; again, while they are unlikely to rank their top alternative last, they may inadvertently swap adjacent alternatives.

Thus, we may wonder if an outcome of a given election would have remained the same if each vote was a perfect reflection of the respective voter's preferences. Of course, the answer to this question depends on the observed election outcome: if the two most successful candidates are close to being tied, it is quite plausible that the error-free outcome would have been different, but if the current winner leads by a significant margin, the election outcome is likely to reflect the true collective opinion. In other words, given an election, it is natural to ask how *robust* its outcome is, given that our perception of the voters' preferences may be noisy.

In this paper, we study this question for several voting rules, namely, the class of all scoring rules and the Condorcet rule, under the assumption that an "elementary" mistake that a voter (or a vote recording device) can make is to swap two adjacent alternatives in the vote; in recording a given vote, several such mistakes can be made consecutively. This approach is motivated by a classic model of noise used in the study of preferences, which is known as the Mallows noise model [7]. However, in contrast to the Mallows model, we do not assume that mistakes follow a particular distribution. Rather, we are interested in the worst-case scenario, i.e., whether the election result could have been different if we were to deviate by δ swaps of adjacent candidates from the observed preference profile. Thus, we measure the distance between elections using the classic *swap distance* [3] (also known as the inversion distance, the bubble-sort distance, or the Kemeny distance), and we ask whether all elections within a given distance bound δ from the observed election E have the same outcome as E.

We remark that this computational problem can be viewed as the destructive version of the well-studied *swap bribery* problem [4] with unit costs. In more detail, in the (constructive version of) the swap bribery problem it is assumed that an external party wants to make a specific candidate the election winner, and bribes the voters to change their preferences; each voter has a price for swapping every pair of candidates in his vote, and the question is whether the external party can achieve its goal given a certain bribery budget. In the destructive version of this problem (which, to the best of our knowledge, has not been considered in the literature), the briber's goal would be to prevent a specific candidate from winning; clearly, this is equivalent to our question under the assumption that all swaps have the same cost.

We are also interested in understanding the structure of elections whose outcome is maximally robust with respect to a given voting rule, i.e., those whose winner is most resilient to swaps of adjacent candidates. Formally for a given voting rule $\mathcal{F}$, we define the *robustness radius* $\mathrm{rob}_{\mathcal{F}}(E, c)$ of an election E with respect to a candidate c as the smallest number of swaps that have to be applied to E to ensure that c is not the (unique) winner of E under $\mathcal{F}$. The *robustness* of a voting rule $\mathcal{F}$ for a given number of voters n and a given number of candidates m is then defined as the maximal robustness radius, over all n-voter m-candidate elections and all candidates in these elections. This quantity measures the maximum resilience of a voting rule to errors in reported preferences and may vary quite substantially from one voting rule to another: for instance, our results show that the Borda rule is considerably more robust than the Condorcet rule.

Our Results We show that our computational problem admits polynomial-time algorithms for all scoring rules and the Condorcet rule. Further, we obtain essentially matching upper and lower bounds on the robustness of several classes of scoring rules, including such prominent scoring rules as Plurality and Borda. Determining the robustness of the Condorcet rule turns out to be more difficult: while we provide non-trivial upper and lower bounds for this quantity, there is still a gap that remains to be closed. Interestingly, we show that an election that is (almost) maximally robust with respect to many scoring rules is provably non-optimal for the Condorcet rule.

Related Work Procaccia et al. [8] also consider robustness of voting rules to swaps of adjacent candidates. However, their approach differs from ours in several important aspects. First, they measure the robustness of a given election as a *fraction* of swaps that leave the outcome unchanged (they also extend this definition to fixed-length chains of swaps), i.e., while our model of noise is adversarial, theirs is random. Second, Procaccia et al. are interested in *minimally* robust elections, while we focus on elections that are *maximally* robust. Indeed, while the goal of Procaccia et al. is to understand which voting rules are most resilient to errors (or, viewed from a different perspective, least sensitive to changes in voters' preferences), and thus a worst-case approach is appropriate, our aim is to understand which features of a preference profile guarantee that a given voting rule will output the desired result, even in the presence of mistakes. Unsurprisingly, our conclusions are also very different from those of Procaccia et al.: in our framework, Borda turns out to be extremely robust, while Plurality is rather fragile, whereas in the model of Procaccia et al. the opposite is true. Finally, we provide efficient algorithms for computing the robustness radius under many voting rules; in contrast, the results of Procaccia et al. are non-algorithmic in nature.

Our work is also closely related to (and shares some of the motivation) with the recent work by Xia [9] on the margin of victory of voting rules. Indeed, Xia explores essentially the same algorithmic question, but for a different model of errors. Namely, he asks if the election results would have remained the same if up to δ voters were to change their vote arbitrarily. Thus, our papers differ in their notion of an elementary error, or, equivalently, in their approach to measuring distance between elections: while the underlying notion of distance for our work is the swap distance, for [9] it is the Hamming distance. In other words, while we study the destructive version of the swap bribery problem [4], paper [9] studies the destructive version of the original bribery problem [5][1].

[1]To be precise, the margin of victory problem studied in [9] differs from destructive bribery with unit costs in its handling of ties, but the two problems are nevertheless very similar; see the discussion in [9].

While our approach is based on a more fine-grained notion of errors than that of [9], we do not claim that it is generally superior: rather, for either approach there is a range of scenarios where it is more suitable than the other. In particular, the swap distance-based model seems more attractive when voters make mistakes due to imperfect introspection or errors in recording their vote, while the Hamming distance-based approach is more appealing when mistakes are due to (potential) malfunctioning of the vote-recording device (which is the motivation put forward in [9]).

We remark that both in our model and in the model of [9] the associated algorithmic question is easy for all scoring rules, but, apart from this, the contribution of the two papers is incomparable: there are several voting rules studied in [9], but not in our work (though we intend to study these voting rules in the future), but, on the other hand, Xia does not consider the Condorcet rule (he does, however, prove NP-hardness results for several voting rules that are refinements of the Condorcet rule). Also, Xia focuses on the algorithmic aspect of the problem only, while a significant (and perhaps the most mathematically interesting) part of our contribution is the study of the combinatorial question of robustness of voting rules; we believe that this question would be just as interesting to study in the model of [9], and propose it as a direction for future work.

The rest of this paper is organized as follows. After introducing our notation and basic definitions in Section 2, we formally define the problems we intend to study (Section 3). Sections 4 and 5 present our results for scoring rules and the Condorcet rule, respectively. We conclude in Section 6.

2 Preliminaries

An *election* is a pair $E = (C, \mathcal{R})$, where C is a set of *candidates*, or *alternatives*, and $\mathcal{R} = (R_1, \dots, R_n)$ is a *preference profile*, with each R_i, $i = 1, \dots, n$, being a linear order over C; we will sometimes write $\succ_i$ in place of R_i. We will refer to the elements of $\mathcal{R}$ as *votes*: R_i is is the vote of the i-th voter in the election $(C, \mathcal{R})$. We denote the number of votes in a preference profile $\mathcal{R}$ by $|\mathcal{R}|$. We say that a voter i prefers $a \in C$ to $b \in C$ if $a \succ_i b$. We denote the candidate ranked by voter i in position j by $c(j, R_i)$. Conversely, we denote the position of a candidate c_j in the i-th vote by $\mathrm{pos}(c_j, R_i)$. We will sometimes identify C with the set $[m] = \{1, \dots, m\}$. We denote the space of all n-voter m-candidate elections by $\mathcal{E}_{n,m}$.

Given an election $E = (C, \mathcal{R})$, a candidate a is said to *win* the pairwise election against b if more than half of the voters prefer a to b; if exactly half of the voters prefer a to b, then a is said to *tie* his pairwise election against b. A candidate $a \in C$ is said to be the *Condorcet winner* of the election $E = (C, \mathcal{R})$ if he beats every other candidate in their pairwise election.

Given two votes R and R' over a set of candidates C, the *swap distance* between R and R', denoted by $d_{\mathrm{swap}}(R, R')$, is the number of swaps of adjacent candidates needed to transform R into R', or, equivalently, the number of pairs $(a, b) \in C \times C$ such that in R candidate a is ranked above candidate b, but in R' candidate b is ranked above candidate a. Given two n-voter elections $E = (C, \mathcal{R})$ and $E' = (C, \mathcal{R}')$ over the same set of candidates C, the *swap distance* between them, denoted by $d_{\mathrm{swap}}(E, E')$, is given by $d_{\mathrm{swap}}(E, E') = \sum_{i=1,\dots,n} d_{\mathrm{swap}}(R_i, R'_i)$.

A *voting correspondence* (in what follows, we will use the terms *voting correspondence* and *voting rule* interchangeably) is a mapping $\mathcal{F}$ that given an election $E = (C, \mathcal{R})$ outputs a non-empty set of candidates $W = \mathcal{F}(E) \subseteq C$; the candidates in W are called the *winners* of the election E under the voting rule $\mathcal{F}$. We will now define the voting rules that will be considered in this paper.

Scoring rules. Every vector of non-negative reals $\alpha = (\alpha_1, \dots, \alpha_m)$ that satisfies $\alpha_1 \geq \dots \geq \alpha_m$ corresponds to a scoring rule $\mathcal{F}_\alpha$, which is defined for m-candidate elections only. Under this rule, each candidate in an election $E = (C, \mathcal{R})$ with $|C| = m$ receives α_i points from every voter that ranks him in position i; the $\mathcal{F}_\alpha$-*score* of a candidate c in E (denoted by $s_\alpha(E, c)$) is the total number of points that c receives in E. The winners under $\mathcal{F}_\alpha$ are the candidates with the highest $\mathcal{F}_\alpha$-score. The vector $(\alpha_1, \dots, \alpha_m)$ is called the *scoring vector* that corresponds to the scoring rule

$\mathcal{F}_\alpha$. As we are interested in asymptotic complexity results, we will consider families of scoring rules $\{\mathcal{F}_{\alpha^m}\}_{m\geq 1}$, where $\alpha^m = (\alpha_1^m, \dots, \alpha_m^m)$ and $\alpha_1^m \geq \cdots \geq \alpha_m^m$. We require these families to be polynomial-time computable, i.e., we assume that for each $m \geq 1$ and each $i = 1, \dots, m$ the number α_i^m is a non-negative integer given in binary, and, moreover, there is a polynomial-time algorithm that can output α_i^m given m and i. There are several prominent voting rules that correspond to families of scoring rules. In particular, Plurality is the family of scoring rules given by $\alpha_1^m = 1$, $\alpha_i^m = 0$ for all $m \geq 1$ and all $i = 2, \dots, m$, Veto is the family of scoring rules given by $\alpha_m^m = 0$, $\alpha_i^m = 1$ for all $m \geq 1$ and all $i = 1, \dots, m-1$, Borda is the family of scoring rules given by $\alpha_i^m = m - i$ for all $m \geq 1$ and all $i = 1, \dots, m$, and k-approval is the family of scoring rules such that for each $m \geq 1$ it holds that $\alpha_i^m = 1$ for $i = 1, \dots, k$ and $\alpha_i^m = 0$ for all $i = k+1, \dots, m$,

The Condorcet rule. Under the Condorcet rule, if the election has a Condorcet winner, he is the (unique) election winner; otherwise, the set of winners is C. We remark that it is more common (see, e.g., [2]) to say that in the latter case the election has no winners. However, in the social choice literature it is standard to require (as we do) that a voting rule outputs a non-empty winner set for every election, so we have modified the definition of the Condorcet rule to satisfy this requirement. Since in this paper we focus on the unique winner variant of our computational problem (see Section 3 for formal definitions), these two definitions are essentially equivalent. However, for the non-unique variant of our problem this is no longer the case; we discuss this issue in detail in Section 5.

In what follows, we abbreviate the Plurality rule to $\mathcal{F}_\mathcal{P}$, the Borda rule to $\mathcal{F}_\mathcal{B}$, k-approval to $\mathcal{F}_k$, and the Condorcet rule to $\mathcal{F}_\mathcal{C}$.

3 Our Model

We will now present the two questions that will be the focus of this paper.

Definition 3.1. *Given a voting rule $\mathcal{F}$, an instance of* $\mathcal{F}$-UC DESTRUCTIVE SWAP BRIBERY *(here "UC" stands for "unit cost") is given by an election $E = (C, \mathcal{R})$, a candidate $c \in C$, and a parameter $\delta \in \mathbb{Z}^+$. It is a "yes"-instance if $\mathcal{F}(E) = \{c\}$, but there exists an election $E' = (C, \mathcal{R}')$ with $d_{\mathrm{swap}}(E, E') \leq \delta$ such that $\mathcal{F}(E') \neq \{c\}$. Otherwise, it is a "no"-instance.*

We remark that in Definition 3.1 we consider the unique winner version of our problem, i.e., we require c to be the unique winner of the original election, and we seek a modified election for which this is no longer the case. Alternatively, one could consider the non-unique winner version of the problem, where c is required to be one of the election winners, and the goal is to find an election in which c is not an election winner at all. It is not hard to verify that the dynamic programming algorithm for scoring rules presented in Section 4 can be modified to work for the non-unique winner version of our problem. However, for the Condorcet rule the relationship between the two variants of the problem is more complicated (see Section 5). We chose to focus on the unique winner version of our problem since it provides a better match for the intuition behind the Condorcet rule.

Definition 3.2. *Given a voting rule $\mathcal{F}$, an election $E = (C, \mathcal{R})$ and a candidate $c \in C$, the* robustness radius *of E with respect to c under $\mathcal{F}$, denoted by $\mathrm{rob}_\mathcal{F}(E, c)$, is the smallest value of δ such that there exists an election $E' = (C, \mathcal{R}')$ with $d_{\mathrm{swap}}(E, E') \leq \delta$ such that $\mathcal{F}(E') \neq \{c\}$.*

Clearly, $\mathrm{rob}_\mathcal{F}(E, c) \geq 0$ and $\mathrm{rob}_\mathcal{F}(E, c) = 0$ if and only if c is not the unique winner of E under $\mathcal{F}$. Moreover, since the swap distance between any pair of n-voter m-candidate elections is at most $\delta_{m,n} = n\frac{m(m-1)}{2}$, we have $\mathrm{rob}_\mathcal{F}(E, c) \leq \delta_{m,n}$ for every $E \in \mathcal{E}_{n,m}$.

Given a voting rule, we would like to understand the structure of the elections that have the maximum robustness radius with respect to this rule. Thus, overloading notation, we define the *robustness of a voting rule $\mathcal{F}$* as a function

$$\mathrm{rob}_\mathcal{F}(m, n) = \max\{\mathrm{rob}_\mathcal{F}(E, c) \mid E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}, c \in C\}.$$

In what follows, we will investigate the complexity of UC DESTRUCTIVE SWAP BRIBERY and prove upper and lower bounds of $\mathrm{rob}_{\mathcal{F}}(m, n)$ for several families of scoring rules as well as the Condorcet rule.

4 Scoring Rules

We start by describing a simple dynamic programming algorithm that efficiently solves UC DESTRUCTIVE SWAP BRIBERY for any polynomial-time computable family of scoring rules. We then describe a simpler and faster algorithm for the Borda rule.

Theorem 4.1. *The problem* $\{\mathcal{F}_{\alpha^m}\}_{m\geq 1}$-UC DESTRUCTIVE SWAP BRIBERY *is in* P *for any polynomial-time computable family of scoring rules* $\{\mathcal{F}_{\alpha^m}\}_{m\geq 1}$.

Proof. Fix a scoring vector $\alpha = (\alpha_1, \dots, \alpha_m)$. We will describe an algorithm that given (a) an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$ that has a unique winner c under $\mathcal{F}_\alpha$ and (b) a positive integer δ, determines whether there exists an election E' with $d_{\text{swap}}(E, E') \leq \delta$ such that $\mathcal{F}_\alpha(E') \neq \{c\}$. The running time of our algorithm will be polynomial in n, m, $\log \delta$ and $\log \alpha_1$. Clearly, this implies the statement of the theorem.

Consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$. Suppose that c is the unique winner of E. For each $a \in C \setminus \{c\}$, we will check whether there exists an election E_a with $d_{\text{swap}}(E, E_a) \leq \delta$ such that in E_a the $\mathcal{F}_\alpha$-score of a is at least as high as that of c; we output “yes” if the answer is positive for at least one $a \in C \setminus \{c\}$. Given an election $E' = (C, \mathcal{R}')$ and a candidate $a \in C \setminus \{c\}$, let $\mathrm{def}(E', a) = \max\{0, s_\alpha(E', c) - s_\alpha(E', a)\}$; we will refer to the quantity $\mathrm{def}(E', a)$ as the *deficit* of a in E'. Thus, our goal is find an election E_a within a distance δ from E such that the deficit of a in E_a is 0.

We start by considering a variant of this problem where we are only allowed to modify a single vote $R_i \in \mathcal{R}$. Suppose that we are allowed to make at most d swaps in R_i. Let $z(i, d)$ be the maximum reduction in a’s deficit that can be obtained in this manner. Clearly, we cannot benefit from swaps that do not involve a or c. Thus, we should use our d swaps to move a upwards or to move c downwards (or both), and it remains to decide how many swaps to allocate to each of these actions; this can be determined by considering all possible splits. More precisely, for each $d' = 0, \dots, d$, we consider the vote $R_i(d')$ obtained by first shifting c by d' positions downwards in R_i and then shifting a by $d - d'$ positions upwards in the resulting vote; among these $d + 1$ votes, we pick one that reduces a’s deficit as much as possible, and let $z(i, d)$ be the corresponding reduction in a’s deficit.

We are now ready to describe the dynamic programming algorithm for our problem. For each $d = 0, \dots, \delta$ and each $i = 0, \dots, n$, let $N(i, d)$ be the smallest deficit of a over all elections at swap distance at most d from E that differ from E in the first i votes only. The quantities $N(i, d)$ can be computed as follows. Clearly, for every $d = 0, \dots, \delta$, $N(0, d)$ is simply a’s deficit in the original election E, which is straightforward to compute. Further, we have

$$N(i, d) = \max\left\{0, \min_{d'=0,\dots,d} (N(i-1, d-d') - z(i, d'))\right\}$$

for all $d = 0, \dots, \delta$ and all $i = 1, \dots, n$. Indeed, we simply have to find an optimal way of splitting d swaps between the i-th vote and the first $i - 1$ votes; the best way to use the d' swaps allocated to the i-th vote is given by $z(i, d')$. Thus, the quantities $N(i, d)$ can be computed inductively starting from $i = 0$. Once we have computed $N(n, \delta)$, it remains to check if $N(n, \delta) = 0$; if yes, we have succeeded in finding an election at distance at most δ from E where a’s score is at least as high as that of c. ☐

For some scoring rules, the algorithm given in the proof of Theorem 4.1 can be simplified. In particular, this is the case for the Borda rule. Indeed, under this rule each upwards swap involving a

but not c, as well as each downwards swap involving c but not a, reduces a's deficit by 1; the most "profitable" swaps are the ones that involve both a and c, as they reduce a's deficit by 2. Thus, our optimal strategy is to maximize the number of "super-profitable" swaps. This observation allows us to simplify our algorithm as follows. We first consider the list $\mathcal{R}' \subseteq \mathcal{R}$ of all votes where c is ranked above a. We re-order the votes in this list according to the number of candidates ranked between c and a, from the smallest to the largest (breaking ties arbitrarily). We then process the votes in $\mathcal{R}'$ one by one. In each vote, we swap c downwards until it is swapped with a. If we have processed all votes in $\mathcal{R}'$, and we still have some swaps available, we allocate them arbitrarily to swapping c downwards or swapping a upwards in any vote in $\mathcal{R}$ where this can be done. Clearly, this approach maximizes the number of swaps that reduce the deficit by 2, and is therefore optimal.

We now move on to the study of robustness of scoring rules. We first provide a simple upper bound that applies to all "reasonable" voting rules. We then show that for the Borda rule this bound is essentially tight.

We say that a voting rule $\mathcal{F}$ is *unanimity-consistent* if in every election E where some candidate c is ranked first by all voters it holds that c is a winner of E under $\mathcal{F}$. Note that all voting rules considered in this paper (and, more broadly, all common voting rules) are unanimity-consistent.

Theorem 4.2. *For any unanimity-consistent voting rule $\mathcal{F}$ we have* $\mathrm{rob}_{\mathcal{F}}(m,n) \leq \frac{nm}{2}$.

Proof. Consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$, and let c be a winner of E under $\mathcal{F}$. For every candidate $a \in C \setminus \{c\}$, let r_a be the number of swaps required to get a into the top position in each vote in $\mathcal{R}$; note that by unanimity consistency performing these r_a swaps would make a an election winner. We have

$$\sum_{a \in C\setminus\{c\}} r_a \leq n(1+2+\ldots+(m-1)) = \frac{nm(m-1)}{2}.$$

As $|C \setminus \{c\}| = m-1$, by the pigeonhole principle there exists some $a \in C \setminus \{c\}$ such that $r_a \leq \frac{nm}{2}$. Hence, $\mathrm{rob}_{\mathcal{F}}(m,n) \leq \frac{nm}{2}$. □

Interestingly, for the Borda rule this bound is essentially tight.

Theorem 4.3. *We have* $\mathrm{rob}_{\mathcal{F}_\mathcal{B}}(m,n) = \frac{nm}{2} + O(n+m)$.

Proof. The upper bound follows immediately from Theorem 4.2. For the lower bound, consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$, where $C = \{c_1, \ldots, c_m\}$ and $\mathcal{R}$ consists of $\lfloor n/2 \rfloor$ votes of the form $c_1 \succ c_2 \succ \ldots \succ c_m$ and $\lceil n/2 \rceil$ votes of the form $c_1 \succ c_m \succ \ldots \succ c_2$. In this election c_1 is the unique Borda winner, and his Borda score is $n(m-1)$. On the other hand, consider a candidate c_i with $i > 1$. His Borda score in E is $(m-i)\lfloor \frac{n}{2} \rfloor + (i-2)\lceil \frac{n}{2} \rceil = \frac{nm}{2} + O(n+m)$.

Now, consider a minimal sequence of swaps that transforms E into an election E' where c_i is a Borda winner. Each swap decreases the difference between the score of c_1 and that of c_i by at most one unless this swap involves both c_1 and c_i (in which case it decreases the difference in their scores by 2); however, there can be at most n swaps of the latter type. Therefore, the total number of swaps required to make c_i an election winner is at least $\frac{nm}{2} + O(n+m)$, and therefore $\mathrm{rob}_{\mathcal{F}_\mathcal{B}}(m,n) \geq \frac{nm}{2} + O(n+m)$. □

Next, we consider the k-approval rule with $k \geq m/2$. We will use the following construction. Given a vote R over a candidate set C of size m, we say that R' is obtained from R by the *downwards shift* if $c(1, R') = c(m, R)$ and for each $j = 2, \ldots, m$ it holds that $c(j, R') = c(j-1, R)$. For instance, by applying the downwards shift to the vote $c_1 \succ \ldots \succ c_{m-1} \succ c_m$ we obtain the vote $c_m \succ c_1 \succ \ldots \succ c_{m-1}$. We say that an election $(C, \mathcal{R}) \in \mathcal{E}_{n,m}$ is an (R, n, m)-*typhoon* if $n = m\alpha$ for some $\alpha \in \mathbb{N}$, $R_1 = R$, for each $i = 2, \ldots, m$ the vote R_i is obtained from the vote R_{i-1} by the downwards shift, and for each $j = 1, \ldots, \alpha - 1$ and each $i = 1, \ldots, m$ it holds that

$R_{mj+i} = R_i$. Further, we say that an election $(C, \mathcal{R}) \in \mathcal{E}_{n,m}$ is a (c, R', n, m)-*lidded typhoon* if $c \in C$, $n = (m-1)\alpha$ for some $\alpha \in \mathbb{N}$, R' is a vote over $C \setminus \{c\}$, and $\mathcal{R}$ is obtained from the $(R', n, m-1)$-typhoon by inserting c into the top position of each vote in $\mathcal{R}'$.

Theorem 4.4. *For $k \geq \frac{m}{2}$ we have $\mathrm{rob}_{\mathcal{F}_k}(m, n) = \frac{n(m-k)^2}{2m} + O(n+m)$.*

Proof. For the upper bound, consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$ that has some candidate c as its unique k-approval winner. Consider a candidate $a \in C \setminus \{c\}$. To ensure that c in not the unique winner of E, it suffices to swap a into the top k positions in each vote. Let r_a denote the number of swaps needed to place a into top k positions in every vote. We have

$$\sum_{a \in C \setminus \{c\}} r_a \leq n(1 + 2 + \ldots + (m-k)) = \frac{n(m-k)(m-k+1)}{2}.$$

As $|C \setminus \{c\}| = m - 1$, by the pigeonhole principle there exists some $a \in C \setminus \{c\}$ such that

$$r_a \leq \frac{n(m-k)(m-k+1)}{2(m-1)} = \frac{n(m-k)^2}{2m} + O(n+m),$$

which establishes our upper bound.

For the lower bound, we provide a proof for the case $n = \alpha(m-1)$ for some $\alpha \in \mathbb{N}$. Our proof can be extended to the case where $m - 1$ does not divide n; we omit the details due to space constraints.

Let R' be a vote over the candidate set $\{c_2, \ldots, c_m\}$ given by $c_2 \succ \ldots \succ c_m$, and let $(C, \mathcal{R})$ be the $(c_1, R', n, m-1)$-lidded typhoon. Clearly, c_1 is the unique winner of $(C, \mathcal{R})$ under k-approval. Fix a candidate c_i with $i > 1$, and consider a minimal sequence of swaps that makes c_i a k-approval winner. Clearly, the only useful swaps are the ones that shift c_1 out of top k positions or ones that shift c_i into top k positions. Shifting c_i into top k positions requires at most $m - k$ swaps, while shifting c_1 out of top k positions requires k swaps, and by our choice of k we have $k \leq m - k$. Thus, an optimal sequence of swaps that makes c_i a k-approval winner is to shift him into top k positions in every vote. Since c_i appears in each of the bottom $m - k$ positions exactly α times, the total number of swaps required is

$$\alpha \frac{(m-k)(m-k+1)}{2} = n \frac{(m-k)(m-k+1)}{2(m-1)} = \frac{n(m-k)^2}{2m} + O(n+m).$$

We conclude that $\mathrm{rob}_{\mathcal{F}_k}(m, n) \geq n(m-k)^2/(2m) + O(n+m)$. □

For k-approval with $k \leq m/2$, the argument in the proof of Theorem 4.4 no longer applies. Specifically, while we conjecture that lidded typhoons are maximally robust for small values of k as well, it is no longer the case that to make some non-top-ranked candidate a an election winner it is optimal to only perform swaps that shift a into the top k positions. Indeed, for small values of k it may be easier to move the top-ranked candidate out of the top k positions. We will now show that this is indeed the case for the Plurality rule.

Theorem 4.5. *For $m \geq 6$, we have $n - 1 - \frac{n}{m-1} \leq \mathrm{rob}_{\mathcal{F}_\mathcal{P}}(m, n) \leq n - \lceil \frac{n}{m-1} \rceil$.*

Proof. For the upper bound, consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$ and suppose that c_1 is the unique Plurality winner of E. Then c_1's Plurality score is at most n. On the other hand, by the pigeonhole principle there exists a candidate $a \in C \setminus \{c\}$ that is ranked in top two positions at least $\lceil \frac{n}{m-1} \rceil$ times. Thus, by using at most $\lceil \frac{n}{m-1} \rceil$ swaps we can ensure that a's Plurality score is at least $\lceil \frac{n}{m-1} \rceil$. Observe that at this point the Plurality score of c is at most $n - \lceil \frac{n}{m-1} \rceil$, so using additional $n - 2\lceil \frac{n}{m-1} \rceil$ swaps, we can reduce its Plurality score to at most $\lceil \frac{n}{m-1} \rceil$. Thus, $\mathrm{rob}_{\mathcal{F}_\mathcal{P}}(m, n) \leq n - \lceil \frac{n}{m-1} \rceil$.

For the lower bound, suppose first that $n = \alpha(m-1)$ for some $\alpha \in \mathbb{N}$. Let $(C, \mathcal{R})$ be the (c_1, R', n, m)-lidded typhoon, where R' is an arbitrary preference order over $C \setminus \{c_1\}$. Among all minimum-length sequences of swaps which ensure that c_1 is not the unique election winner under Plurality, pick one which swaps c_1 out of the top position in the maximum number of votes, and let c_i, $i > 1$, be a winner of the resulting election E'. Let N_1 be the set of voters in E' that rank c_1 first, let N_i be the set of voters in E' that rank c_i first, and let $N' = N \setminus (N_c \cup N_i)$ be the set of all other voters; we have $|N_i| \geq |N_1|$.

We have $N' \neq \emptyset$, since otherwise we would have $|N_i| \geq n/2$, and for $m \geq 6$ the cost of swapping c_i into the top position in $n/2$ votes exceeds n. Therefore, we have $|N_i| = |N_1|$. Indeed, if $|N_i| > |N_1|$, we could shorten our swap sequence by not making the swaps in some vote in N': in the resulting election it would still be the case that $|N_i| \geq |N_1|$. Now, suppose that $|N_i| > \alpha$. Then we had to perform at least two swaps in at least one vote in N_i. Consider a modified sequence of swaps that performs no swaps in this vote (so that it still ranks c_1 first), but swaps c_1 out of the top position in two votes in N_1. The length of this modified sequence is at most that of the original sequence, it also ensures that c_i's Plurality score is at least as high as that of c_1, and it swaps c_1 out of the top position in a higher number of votes, a contradiction with our choice of the swap sequence. It follows that $|N_i| = |N_1| = \alpha$, which implies that the length of our swap sequence is at least $n - \alpha = n - \frac{n}{m-1}$.

It is easy to generalize this argument to the case where $m - 1$ does not divide n to obtain a slightly weaker lower bound of $n - 1 - \frac{n}{m-1}$; we omit the details. □

It is instructive to compare the bounds obtained in Theorems 4.3, 4.4, and 4.5. Perhaps not surprisingly, among all k-approval rules with $k \geq m/2$, the $m/2$-approval rule is the most robust, and Veto is the least robust. It is interesting to note that Borda is about four times more robust that $m/2$-approval and $m/2$ times more robust than Plurality; also Plurality is considerably more robust than Veto.

5 The Condorcet Rule

In this section, we show that UC DESTRUCTIVE SWAP BRIBERY remains easy for the Condorcet rule; however, deriving good bounds on $\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(m, n)$ requires quite a bit of effort.

Theorem 5.1. *The problem* $\mathcal{F}_{\mathcal{C}}$-UC DESTRUCTIVE SWAP BRIBERY *is in* P.

Proof. Consider an instance of $\mathcal{F}_{\mathcal{C}}$-UC DESTRUCTIVE SWAP BRIBERY given by an election $E = (C, \mathcal{R})$, a candidate $c \in C$ and a non-negative integer δ. Suppose that c is the Condorcet winner of E. Similarly to the proof of Theorem 4.1, for every candidate $a \in C \setminus \{c\}$ we check if there exists an election E_a with $d_{\mathrm{swap}}(E, E_a) \leq \delta$ such that a beats or ties c in their pairwise election. It is not hard to see that we can use essentially the same algorithm as for the Borda rule: that is, we order the votes where a is ranked below c according to the distance between c and a (from the smallest to the largest) and process these votes one by one, shifting c downwards to appear just below a; we do this until we exhaust our swap budget. We return "yes" if in the end of this process a beats or ties c in their pairwise election. □

We remark that the proof of Theorem 5.1 does not extend to the the co-winner version of the $\mathcal{F}_{\mathcal{C}}$-UC DESTRUCTIVE SWAP BRIBERY problem. Indeed, suppose that c is a co-winner of an election E. Then the nearest election where c is not a co-winner is one where some other candidate is the (unique) Condorcet winner. Thus, given an election E with no Condorcet winners (where, according to our definition of the Condorcet rule, all candidates are the election winners), solving the co-winner version of $\mathcal{F}_{\mathcal{C}}$-UC DESTRUCTIVE SWAP BRIBERY is essentially the problem of computing the winners of E under the Dodgson rule (recall that under this rule, the winners are the candidates who can be made the Condorcet winners by the smallest number of swaps of adjacent candidates).

The latter problem is known to be computationally hard [1, 6]. In fact, we can use the results of [1, 6] to show that the co-winner version of $\mathcal{F}_{\mathcal{C}}$-UC DESTRUCTIVE SWAP BRIBERY is computationally hard as well. We do not present the formal proof of this result, as we do not find the co-winner version of $\mathcal{F}_{\mathcal{C}}$-UC DESTRUCTIVE SWAP BRIBERY intuitively appealing, and therefore we do not think that this hardness result is informative.

We will now present our upper and lower bounds on the robustness of the Condorcet rule.

It will be convenient to prove bounds on $\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(m+1,n)$ rather than $\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(m,n)$; our results are not affected by this change, since they involve an error term than is linear in $n+m$. First, we will restate the problem of computing $\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(m+1,n)$ as an optimization problem. Given a set $S \subseteq \mathbb{N}$, let $L(S)$ denote the sum of the smallest $\lceil\frac{|S|}{2}\rceil$ numbers in S. Then, given an election $(C,\mathcal{R})$ with $|\mathcal{R}|=n$, the quantity $L(\{\mathrm{pos}(c,R_i) \mid i \in [n]\})$ is the sum of the lowest $\lceil n/2 \rceil$ positions in which candidate c appears in $\mathcal{R}$. We can now reformulate our problem as follows.

Lemma 5.2. *We have* $\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(m+1,n) = \max_{(C,\mathcal{R})\in\mathcal{E}_{n,m}} \min_{c\in C} L\left(\{\mathrm{pos}(c,R_i) \mid i\in[n]\}\right)$.

Proof. The proof of Theorem 5.1 shows that for every election $E'=(C',\mathcal{R}') \in \mathcal{E}_{n,m+1}$ and every $c_j \in C'$ we have

$$\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(E',c_j) = \min_{c\neq c_j} L\left(\{\max\{0, \mathrm{pos}(c,R'_i) - \mathrm{pos}(c_j,R'_i)\} \mid i\in[n]\}\right).$$

Indeed, to ensure that c_j is not the unique winner of E' under the Condorcet rule, we need to make c_j tie with or lose to some other candidate $c \neq c_j$, i.e., c has to be ranked higher than c_j in at least $\lceil n/2\rceil$ votes. For each $c \in C'\setminus\{c_j\}$, the number of swaps needed to make c appear above c_j in vote i is $\max\{0,\mathrm{pos}(c,R'_i)-\mathrm{pos}(c_j,R'_i)\}$, and to minimize the total number of swaps for c, we take the $\lceil n/2\rceil$ votes for which we need the smallest number of swaps. Finally, we choose a candidate $c\in C'\setminus\{c_j\}$ for which the required number of swaps is the smallest.

Now, consider an election $E'=(C',\mathcal{R}')\in\mathcal{E}_{n,m+1}$ and a candidate $c_j\in C'$. Let $E^j = (C',\mathcal{R}^j)$ be the election obtained by moving c_j to the top of each vote in E' (and not changing the relative order of the remaining candidates). We can simplify the expression for $\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(E^j,c_j)$, since we have $\mathrm{pos}(c_j,R^j_i)=1$ and $\mathrm{pos}(c,R^j_i) > \mathrm{pos}(c_j,R^j_i)$ for all $i\in[n]$ and all $c\in C'\setminus\{c_j\}$. Thus, we obtain

$$\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(E^j,c_j) = \min_{c\neq c_j} L(\{\mathrm{pos}(c,R^j_i)-1 \mid i\in[n]\}).$$

On the other hand, it is not hard to see that $\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(E',c_j) \le \mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(E^j,c_j)$. Thus, when computing $\mathrm{rob}_{\mathcal{F}_{\mathcal{C}}}(m+1,n)$, we only need to consider elections where some candidate c_j is ranked first in every vote; denote the set of all such elections by $\mathcal{E}^j_{m+1,n}$. Note also that the identity of this candidate does not matter. Now, take an election $E^j=(C',\mathcal{R}^j)\in\mathcal{E}^j_{m+1,n}$, let $C=C'\setminus\{c_j\}$ and consider an election $E=(C,\mathcal{R})\in\mathcal{E}_{n,m}$ obtained by removing c_j from each vote in E^j. Note that any election over C can be obtained in this way.

For every $c\in C$ we have $L(\{\mathrm{pos}(c,R_i) \mid i\in[n]\}) = L(\{\mathrm{pos}(c,R^j_i)-1 \mid i\in[n]\})$. Consequently,

$$\mathrm{rob}_{\mathcal{F}}(m+1,n) = \max_{(C,\mathcal{R})\in\mathcal{E}_{n,m}} \min_{c\in C} L(\{\mathrm{pos}(c,R_i) \mid i\in[n]\}). \qquad \square$$

From now on, to simplify notation, we identify the candidate set C with $[m]$ and let $s_j = L(\{\mathrm{pos}(j,R_i) \mid i\in[n]\})$ for each candidate $j\in[m]$. By Lemma 5.2, it suffices to find upper and lower bounds on $\max_{E\in\mathcal{E}_{n,m}}\min_{j\in[m]} s_j$. The next theorem provides a lower bound.

Theorem 5.3. *For every $m,n\in\mathbb{N}$ there exists an election $E=(C,\mathcal{R})\in\mathcal{E}_{n,m}$ such that $s_j \ge \frac{1}{6}mn + O(m+n)$ for every candidate $j\in[m]$.*

Proof. We start by giving the proof for the case $m = 3k$, $n = 6\ell$ for some $k, \ell \in \mathbb{N}$.

For each $j = 1, \dots, k$, we place the candidates j, $m - 2j + 1$, and $m - 2j + 2$ in positions j, $m - 2j + 1$, and $m - 2j + 2$ in each vote so that each of them appears 2ℓ times in each position:

$$\begin{array}{rccc} j: & j \quad \dots \quad j & m-2j+1\dots m-2j+1 & m-2j+2\dots m-2j+2 \\ m-2j+1: & m-2j+2\dots m-2j+2 & j \quad\dots\quad j & m-2j+1\dots m-2j+1 \\ m-2j+2: & \underbrace{m-2j+1\dots m-2j+1}_{2\ell} & \underbrace{m-2j+2\dots m-2j+2}_{2\ell} & \underbrace{j \quad \dots \quad j}_{2\ell} \end{array}$$

Clearly, this results in a valid profile over $[m]$. For instance, for $m = n = 6$ we obtain the following profile:

$$\begin{pmatrix} 1 & 1 & 5 & 5 & 6 & 6 \\ 2 & 2 & 3 & 3 & 4 & 4 \\ 4 & 4 & 2 & 2 & 3 & 3 \\ 3 & 3 & 4 & 4 & 2 & 2 \\ 6 & 6 & 1 & 1 & 5 & 5 \\ 5 & 5 & 6 & 6 & 1 & 1 \end{pmatrix}$$

In such an election, for every $j \in \{1, \dots, k\}$ we have

$$s_j = j \times 2\ell + (m - 2j + 1) \times \ell = m\ell + \ell = \frac{1}{6}mn + O(m + n).$$

By symmetry, $s_j = s_{m-2j+1} = s_{m-2j+2}$. Therefore, $s_j = \frac{1}{6}mn + O(m + n)$ for all $j \in [m]$.

We will now consider the general case, i.e., we drop the assumption that m is divisible by 3 and n is divisible by 6. First, we fill in the top $3\lfloor \frac{m}{3} \rfloor$ rows and the first $6\lfloor \frac{n}{6} \rfloor$ columns of the profile with $3\lfloor \frac{m}{3} \rfloor$ candidates as described above. Then we complete each of these $6\lfloor \frac{n}{6} \rfloor$ columns by an arbitrary permutation of the remaining candidates. Each remaining column can be an arbitrary vote over $[m]$. It is not difficult to adapt the proof for the special case $m = 3k$, $n = 6\ell$ to show that the theorem holds for this profile. □

Combining Theorem 5.3 with Lemma 5.2, we obtain $\mathrm{rob}_{\mathcal{F}}(m + 1, n) \geq \frac{1}{6}mn + O(m + n)$ and hence

$$\mathrm{rob}_{\mathcal{F}}(m, n) \geq \frac{1}{6}(m - 1)n + O(m + n) = \frac{1}{6}mn + O(m + n).$$

Now we consider the upper bound.

Theorem 5.4. *For any $E \in \mathcal{E}_{n,m}$ there exists a candidate j such that $s_j \leq \lambda mn + O(m + n)$ for any constant $\lambda > (\sqrt{3} - 1)/4$.*

Proof. Fix $\lambda > (\sqrt{3} - 1)/4$ and suppose for the sake of contradiction that $s_j > \lambda mn + O(m + n)$ for each $j \in [m]$. Given an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$, we construct an $m \times n$ matrix $M(\mathcal{R})$ as follows. The j-th row of $M(\mathcal{R})$ lists all n positions in which candidate j occurs in the n votes, in non-decreasing order. Below is an example of a 3×4 profile $\mathcal{R}$ and its corresponding matrix $M(\mathcal{R})$.

$$\mathcal{R} = \begin{pmatrix} 1 & 2 & 3 & 3 \\ 2 & 3 & 2 & 2 \\ 3 & 1 & 1 & 1 \end{pmatrix} \qquad M(\mathcal{R}) = \begin{pmatrix} 1 & 3 & 3 & 3 \\ 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 3 \end{pmatrix}$$

By the definition of $M(\mathcal{R})$, each number between 1 and m (which denotes a position in a vote) appears exactly n times in $M(\mathcal{R})$. Moreover, s_j is simply the sum of the leftmost $\ell = \lceil \frac{n}{2} \rceil$ entries of the j-th row in $M(\mathcal{R})$. Let S denote the submatrix formed by the first ℓ columns of $M(\mathcal{R})$, and let Σ denote the sum of all entries of S. We will derive upper and lower bounds on Σ. For $\lambda > (\sqrt{3}-1)/4$ the lower bound will exceed the upper bound, leading to a contradiction.

As we have assumed that $s_j > \lambda mn + O(m+n)$, a lower bound is immediate:

$$\Sigma = \sum_{i=1}^{m} s_j > \lambda m^2 n + O(m^2 + mn).$$

The upper bound requires much more work. Let a be the smallest entry of the ℓ-th column of $M(\mathcal{R})$, and let i_0 be the index of its row. All entries to the left of a do not exceed a, so $s_{i_0} \leq \ell a$. On the other hand, our assumption implies $s_{i_0} > \lambda mn + O(m+n)$, so we get a lower bound on a: $a > 2\lambda m + O(\frac{m+n}{n})$.

Note that each entry of $M(\mathcal{R})$ that is not in S is at least a. Therefore, all entries that are smaller than a have to appear in S, and each number between 1 and $a-1$ has to appear exactly n times. The sum of these numbers is

$$\Sigma_1 = \sum_{i=1}^{a-1} i \cdot n = \frac{1}{2} a^2 n + O(mn).$$

Let $\Sigma_2 = \Sigma - \Sigma_1$; Σ_2 is the sum of all entries of S that are greater than or equal to a. We will now derive an upper bound on Σ_2, which will imply an upper bound on Σ.

Let $N_{\geq k}$ denote the number of entries in S that are greater than or equal to k. We will first obtain a general upper bound on $N_{\geq k}$. Observe that entries with value k appear in at least $\lceil \frac{N_{\geq k}}{\ell} \rceil$ rows, and each entry in these rows that does not appears in S is greater than or equal to k. Hence the total number of entries that are greater than or equal to k is at least $N_{\geq k}$ (in S) plus $(n-\ell)\lceil \frac{N_{\geq k}}{\ell} \rceil$ (not in S). On the other hand, there are exactly $(m-k+1)n$ entries that are greater than or equal to k, so we get

$$N_{\geq k} \leq \frac{(m-k+1)n}{1 + \frac{n-\ell}{\ell}} = (m-k+1)\ell.$$

In total there are $m\ell$ entries in S, which include the $n(a-1)$ entries that are smaller than a. We want an upper bound for the sum of the remaining $m\ell - n(a-1)$ entries. To maximize Σ_2, the best way to fill up the remaining entries is to set $N_{\geq k} = (m-k+1)\ell$ by using entries $k = m, m-1, \ldots$ until we run out of entries. More specifically, we put in ℓ entries of value $m, m-1, \ldots, 2a-1$, respectively, and after that the entries left are negligible, since there are at most $a-1$ of them (as $\ell \leq (n+1)/2$) and the order of their sum is $O(m^2)$). Therefore,

$$\Sigma_2 \leq \sum_{i=2a-1}^{m} i \cdot \ell + O(m^2) = \frac{1}{2}(m+2a-1)(m-2a+2)\ell + O(m^2) = \frac{1}{2}(m^2 - 4a^2)\frac{n}{2} + O(m^2 + mn).$$

Combining Σ_1 and Σ_2, we obtain

$$\Sigma = \Sigma_1 + \Sigma_2 \leq \frac{1}{4}(2a^2 + m^2 - 4a^2)n + O(m^2 + mn) = \frac{1}{4}(m^2 - 2a^2)n + O(m^2 + mn),$$

which, by the lower bound on a, can be upper-bounded as

$$\frac{1}{4}(m^2 - 2 \cdot 4\lambda^2 m^2)n + O(m^2 + mn) = \frac{1}{4}(1 - 8\lambda^2)m^2 n + O(m^2 + mn).$$

The lower bound on Σ exceeds this upper bound when $\lambda m^2 n > \frac{1}{4}(1 - 8\lambda^2)m^2 n$, i.e., $8\lambda^2 + 4\lambda - 1 > 0$, which holds for $\lambda > (\sqrt{3} - 1)/4$.

□

Combining Theorem 5.4 with Lemma 5.2, we obtain $\mathrm{rob}_{\mathcal{F}_C}(m+1, n) \leq \lambda mn + O(m+n)$ and hence $\mathrm{rob}_{\mathcal{F}_C}(m, n) \leq \lambda(m-1)n + O(m+n) = \lambda mn + O(m+n)$ for every $\lambda > (\sqrt{3} - 1)/4$. Thus, we have

$$\frac{mn}{6} + O(m+n) \leq \mathrm{rob}_{\mathcal{F}_C}(m, n) \leq (\frac{\sqrt{3} - 1}{4} + \varepsilon)mn + O(m+n)$$

for every $\varepsilon > 0$. We have $1/6 \approx 0.167$ and $(\sqrt{3}-1)/4 \approx 0.183$, i.e., there is a small gap between our lower and upper bounds. Closing this gap is a natural direction for future work. We remark that our bounds indicate than the Condorcet rule is considerably less robust than the Borda rule, but more robust than $m/2$-approval. Also, it is interesting to note that the lidded typhoon is *not* the most robust election with respect to the Condorcet rule.

6 Conclusions and Future Work

We have introduced the notions of robustness radius of an election and robustness of a voting rule. We have provided efficient algorithms for computing the robustness radius of a given election with respect to scoring rules and the Condorcet rule, and we have provided bounds on the robustness of several voting rules, including Plurality, Borda, k-approval for $k \geq m/2$ and the Condorcet rule. It would be interesting to see if our algorithmic results for destructive swap bribery can be extended to voting rules not considered in this paper (such as, e.g., Copeland and Maximin) and to the general cost version of this problem. Similarly, a natural research direction would be to analyze the robustness of other voting rules.

We remark that the robustness notions introduced in this paper are defined in terms of the swap distance. However, one can define and study them for other distances over elections, such as the Hamming distance or the footrule distance. In particular, one might be able to use the techniques developed by Xia [9] in order to study robustness of voting rules with respect to the Hamming distance.

Acknowledgments This research was supported by National Research Foundation (Singapore) under grant 2009-08. We thank Arkadii Slinko and Piotr Faliszewski for useful discussions.

References

[1] J. Bartholdi, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

[2] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[3] M. M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, 2009.

[4] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *SAGT'09*, pages 299–310, 2009.

[5] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of AI Research*, 35:485–532, 2009.

[6] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[7] C. L. Mallows. Non-null ranking models. *Biometrica*, 44:114–130, 1957.

[8] A. D. Procaccia, J. S. Rosenschein, and G. A. Kaminka. On the robustness of preference aggregation in noisy environments. In *AAMAS'07*, pages 416–422, 2007.

[9] L. Xia. Computing the margin of victory for various voting rules. In *ACM EC'12*, 2012.

Dmitry Shiryaev, Lan Yu, Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
Email: SHIR0010@ntu.edu.sg, YULA0001@ntu.edu.sg, eelkind@ntu.edu.sg

Goodness of fit measures for revealed preference tests: Complexity results and algorithms

Laurens Cherchye, Bram De Rock, Bart Smeulders and Frits C.R. Spieksma

Abstract

We provide results on the computational complexity of goodness of fit measures (i.e. Afriat's efficiency index, Varian's efficiency vector-index and the Houtman-Maks index) associated with several revealed preference axioms (i.e. WARP, SARP, GARP and HARP). Our NP-Hardness results are obtained by reductions from the independent set problem. We also show that this reduction can be used to prove that no constant factor approximations algorithm exists for the Houtman-Maks index (unless P = NP). Finally, we give an exact polynomial time algorithm for finding Afriat's efficiency index.

Keywords: Revealed preference, Complexity, Nonparametric rationality tests

1 Introduction

Utility maximization is a core hypothesis in neoclassical microeconomics, and testing the empirical validity of this assumption has attracted considerable attention in the literature. Such tests based on revealed preference theory have become increasingly popular. An attractive feature of these tests is that they are intrinsically nonparametric: they check consistency with the utility maximization hypothesis without requiring a (typically nonverifiable) functional specification of the utility function; and so they maximally avoid the risk of erroneous conclusions due to a misspecified functional form. The empirical requirements for utility maximization are summarized in terms of revealed preference axioms, which can be directly applied to consumption data (prices and quantities) without requiring auxiliary assumptions. For example, a key result of revealed preference theory is that consumption can be represented as maximizing a (well-behaved) utility function if and only if it satisfies the Generalized Axiom of Revealed Preference (GARP) [26]. Three other axioms that are most frequently considered in the applied literature are the Weak, Strong and Homothetic Axioms of Revealed Preference (WARP, SARP and HARP; see Section 2 for exact definitions).

However, a frequently cited weakness of the basic revealed preference tests is that they are 'sharp' tests: they only tell us whether or not observed behavior is exactly consistent with the revealed preference axiom that is being tested. When consumption data do not pass the test, there is no indication concerning the severity or the amount of violations. To deal with this, a number of measures have been proposed in the literature to express how close a data set is to satisfying rationality. In what follows, we will call these measures "goodness of fit" measures; they tell us how well a revealed preference axiom fits the data at hand. Probably the most popular goodness of fit measure in applied work is Afriat's efficiency index (AI) [1]. Other frequently used measures are the ones of Houtman and Maks (HI) [15] and Varian (VI) [27]. Section 2 provides a precise description of these alternative goodness of fit measures.

The revealed preference axioms and goodness of fit measures have been used intensively in the applied literature. The first tests of the axioms of revealed preference go back to the sixties and seventies. Aggregated household consumption data was used in tests of SARP by Koo [18, 19], Koo and Hasenkamp [20], Mossin [24] and Landsburg [21]. Varian [26] tested GARP using similar data. Only Koo tried to measure the severity of the rejections by focusing on the amount of violations and using a measure similar to HI. Over the last decade, the goodness of fit measures have been used more and more often. Sippel [25] tests relaxations of WARP, SARP and GARP related to AI. AI and GARP are used in papers by Mattei [23], Harbaugh et al. [14] Andreoni and Miller [4], Février and Visser [13], Choi et al. [7, 8], Dean and Martin [11] and Burghart [6]; the last four papers also use HI. VI and GARP appears in Cox [10], Mattei [23], Choi et al. [7, 8] and Dean and Martin [11]. For WARP, all three indices appear in Choi et al. [7]. To the best of our knowledge, there do not exist any studies that compute goodness of fit measures for HARP, although there exist papers in which HARP is tested (see for example Manser and McDonald [22]). Finally, we also note continuing interest in goodness of fit measures, illustrated by the recent introduction of several new indices in the literature; specifically the money pump index by Echenique et al. [12] which calculates the monetary cost of irrational behaviour and the minimal swaps and minimal loss index by Apesteguia and Ballester [5].

This paper is specifically concerned with the computational complexity of the goodness of fit measures used in revealed preference analysis. In general, computational complexity becomes an important issue if one wants to consider large data sets. In this respect, we indicate that large consumption data sets are increasingly available (see e.g. the scanner consumption data that nowadays can be used), which directly motivates the research question we consider here. Indeed, while the computational complexity of methods for testing GARP and the other revealed preference axioms is well understood by now, this is not always the case for computing the above mentioned goodness of fit measures.

It is generally thought that calculating AI is easy. However, to our knowledge, no exact algorithm exists in the literature. Varian [27] provides an approximation algorithm, which comes within $\left(\frac{1}{2}\right)^m$ of the true index-value in m GARP tests. As for the other two indices (HI and VI), it has been empirically recognized that computing them is computationally intensive.[1] For instance, Varian [27] writes:

> "Computing the set of efficiency indices [VI] that are as close as possible to 1 in some norm is substantially more difficult ... This approach is significantly more difficult from a computational perspective."

Similarly, Choi et al. [8] state:

> "All indices [VI and HI] are computationally intensive for even moderately large data sets."

The goal of the current paper is to give a theoretical foundation for these practical observations and to strengthen the existing results. As far as we are aware, explicit complexity results are known only for index HI. More specifically, Houtman and Maks establish a link between their index for SARP and feedback vertex set on a digraph, which implies NP-Hardness. Next, Dean and Martin [11] state that HI for GARP is also NP-HARD.

We define the computational complexity for every combination of the three goodness of fit measures (AI, VI and HI) and the four revealed preference axioms (GARP, SARP, WARP and HARP) mentioned above. We will refer to these problems as $\{A, V, H\}$I-$\{G, S, W, H\}$ARP, where choosing a symbol from the set $\{A, V, H\}$ and a symbol from the set $\{G, S, W, H\}$ identifies a particular problem. For example, AI-GARP is the problem of computing the maximum index AI such that the data set satisfies a relaxation of GARP.

[1] Because of the difficulty to exactly calculate VI, some authors have focused on designing approximate heuristics. See, for example, Varian [28] and Alcantud et al. [3].

Our main results are summarized in Table 1, where a column corresponds to a specific axiom and a row to a specific measure and where n stands for the number of observations.[2]

	WARP	SARP	GARP	HARP
AI (sec 6)	$n^2 \log n$	$n^2 \log n$	$n^{2.376} \log n$	n^3
VI (sec 4)	NP-HARD	NP-HARD	NP-HARD	NP-HARD
HI (sec 5)	Inapproximable	Inapproximable	Inapproximable	Inapproximable

Table 1: Overview of Results

The rest of this paper unfolds as follows. The next section sets the stage by introducing the basic revealed preference concepts that we will use throughout. Section 3 provides a statement of the computational problems we focus on. Sections 4 and 5 then presents our results on computational complexity for the indices VI and HI. Section 6 does the same for the index AI. Here, we also give exact polynomial time algorithms for computing this index in practical applications. Section 7 concludes.

2 Revealed preference concepts

We start by stating the 4 revealed preference axioms that we will consider. Subsequently, we present the different goodness of fit measures.

2.1 Axioms of Revealed Preference

Our analysis starts from a data set $S = \{(p_i, q_i)|\ i = 1, \ldots, n\}$, where p_i (q_i) is an N-dimensional vector of prices (quantities) corresponding to observation $i = 1, \ldots, n$. Without loss of generality, we will assume that prices are normalized such that $p_i q_i = 1$ for every observation i.

To define the concept of revealed preferences we consider two observations i and j. If $(p_i q_i =)\ 1 \geq p_i q_j$, we say that bundle q_i is *directly revealed preferred* to bundle q_j. This is expressed by writing $q_i R_0 q_j$, where R_0 captures the direct revealed preference relation. The transitive closure of R_0 is denoted by R and is called the indirect revealed preference relation. If $1 > p_i q_j$, we say that bundle q_i is *strictly directly revealed preferred* to bundle q_j, which is denoted by $q_i P_0 q_j$. Finally, P stands for the transitive closure of P_0.

We can then state the four revealed preference axioms that we consider in this paper.

Definition 1. ***(WARP)*** *A data set S satisfies WARP if for each pair of bundles, q_i, q_j $(i, j = 1, \ldots, n$ with $i \neq j)$, the following holds: if $q_i R_0 q_j$ then it is not the case that $q_j R_0 q_i$.*

Definition 2. ***(SARP)*** *A data set S satisfies SARP if for each pair of bundles, q_i, q_j, $(i, j = 1, \ldots, n$ with $i \neq j)$, the following holds: if $q_i R q_j$ then it is not the case that $q_j R_0 q_i$.*

Definition 3. ***(GARP)****: A data set S satisfies GARP if for each pair of bundles, q_i, q_j, $(i, j = 1, \ldots, n$ with $i \neq j)$, the following holds: if $q_i R q_j$ then it is not the case that $q_j P_0 q_i$.*

Definition 4. ***(HARP)****: A data set S satisfies HARP if for every sequence of observations, $i, j, k, \ldots, l (= 1, \ldots, n)$, the following holds:* $\log(p_i q_j) + \log(p_j q_k) + \ldots + \log(p_l q_i) \geq 0$.

In words, the main differences between the alternative axioms can be summarized as follows (see Varian [29] for a more extensive discussion on the meaning of the axioms). Data consistency with WARP is a necessary condition for data consistency with SARP;

[2] *'inapproximable'* stands for: no polynomial-time algorithm can achieve a constant-factor approximation unless $P = NP$.

the essential difference is that WARP (in contrast to SARP) does not require transitivity of preferences. Next, data consistency with SARP means that consumption behavior can be described as maximizing a utility function that generates single-valued demand. Similarly, data consistency with GARP means that consumption behavior can be described as maximizing a utility function that generates multi-valued demand. As such, GARP is a generalization of SARP. Finally, data consistency with HARP means that consumption behavior can be described as maximizing a utility function that is homothetic. This implies that GARP is a necessary condition for HARP.

2.2 Goodness of fit measures

In practice, direct application of any of the above revealed preference axioms to some given data set effectively obtains a 'sharp' test: a data set either satisfies the axiom or it does not. In words, such a test allows us to conclude whether or not observed behavior is 'exactly' consistent with the hypothesis of utility maximization (of a particular form, depending on whether we consider WARP, SARP, GARP or HARP). However, a data set that is not exactly consistent may actually be very close to consistency. For example, there may be only a limited number of observations that cause the observed violations of the axiom that is subject to testing. Or, the violations may be very insignificant in that small adjustments of the observations' expenditures (i.e. prices times quantities) may suffice to obtain consistency. Generally, it is interesting to quantify the degree to which a given data set is close to consistency (see [27] for extensive motivation).

To account for these considerations, a number of goodness of fit measures have been described in the literature. Three often used measures are Afriat's efficiency index (AI), Varian's efficiency vector index (VI) and the Houtman and Maks index (HI). Essentially, the indices AI and VI look for minimal expenditure perturbations to obtain consistency with the revealed preference axiom under evaluation: the AI index applies a common perturbation to all observations, while the VI index allows a different perturbation for each individual observation. Next, the index HI identifies the largest subset of observations that are consistent with the axiom. Essentially, this quantifies the degree of violation in terms of the number of observations that are involved in a violation of the revealed preference axiom that is tested. We refer to Varian [29] for a more detailed discussion of the different goodness of fit measures we evaluate.

To formally introduce our goodness of fit measures, we make use of the vector $e = (e_1, e_2, \ldots, e_n)$, with $0 \leq e_i \leq 1$. This vector introduces an index e_i for each observation i, which relaxes the revealed preference relations R_0 and P_0 as follows:

$$\text{if } e_i(= e_i p_i q_i) \geq p_i q_j \text{ then } q_i R_0(e) q_j,$$

$$\text{if } e_i(= e_i p_i q_i) > p_i q_j \text{ then } q_i P_0(e) q_j.$$

Analogous to before, $R(e)$ and $P(e)$ represent the transtive closures of $R_0(e)$ and $P_0(e)$. These newly defined relations $R_0(e)$, $P_0(e)$, $R(e)$ and $P(e)$ give rise to relaxed versions of the earlier axioms of revealed preference, which are defined for a given vector e. Clearly these axioms comply with the original versions of WARP, SARP, GARP and HARP as soon as $e_i = 1$ for all i.

Definition 5. ***(WARP(e))*** *A data set S satisfies WARP(e) if for each pair of bundles, $(i, j = 1, \ldots, n$ with $i \neq j)$, the following holds: if $q_i R_0(e) q_j$ then it is not the case that $q_j R_0(e) q_i$.*

Definition 6. ***(SARP(e))*** *A data set S satisfies SARP(e) if for each pair of bundles, q_i, q_j, $(i, j = 1, \ldots, n$ with $i \neq j)$, the following holds: if $q_i R(e) q_j$ then it is not the case that $q_j R_0(e) q_i$.*

Definition 7. ***(GARP(e))*** *A data set S satisfies GARP(e) if for each pair of bundles,* q_i, q_j*,* $(i, j = 1, \ldots, n$ *with* $i \neq j)$*, the following holds: if* $q_i R(e) q_j$ *then it is not the case that* $q_j P_0(e) q_i$.

Definition 8. ***(HARP(e))*** *A data set S satisfies HARP(e) if for every sequence of observations* $i, j, k, \ldots, l(= 1, \ldots, n)$*,, the following holds:* $\log(p_i q_j) + \log(p_j q_k) + \ldots + \log(p_l q_i) \geq \log(e_i) + \log(e_j) + \ldots + \log(e_l)$.

To define the Afriat Index (AI), we assume that $e_1 = \cdots = e_n$, which does indeed comply with a common perturbation for all observations. The index AI equals the highest value for which the data is consistent with the tested revealed preference axiom. More precisely, if $AI = 1$, then the data is consistent with the tested axiom. While if $AI < 1$, then this indicates that we need to pertubate the data to make it consistent with the revealed preference axiom under study. The smaller the number AI is, the higher the perturbation or, alternatively, the more severe the rejection of the axiom. Finally, we note that AI is well-defined. If for a given e the data is consistent with, for example, WARP(e), then the same holds for all $e' < e$. Indeed, by construction we have that the revealed preference relations in terms of e' are always a subset of the ones in terms of e (e.g. $R_0(e') \subseteq R_0(e)$).

The Varian Index (VI) differs from the index AI by allowing for observation specific perturbations. The index VI equals the vector e that is closest to one, for some given norm, such that the data satisfies the revealed preference axiom under study. For example, if we use the quadratic norm, then VI should minimize $\sum_i (1 - e_i)^2$ such that, for example, WARP(e) is satisfied. Further, the index VI is subject to the same qualifications as the index AI.

Finally, the Houtman and Maks index (HI) equals the size of the largest subset of observations which satisfy the axioms of revealed preference. Formally, this complies with restricting the possible values of e_i so that $e_i \in \{0, 1\}$.

3 Problem statement

In this section we introuduce the tools that we need to prove the results announced in Table 1. In particular, in Section 3.1 we show how to reformulate the goodness-of-fit measures using graph theory and in Section 3.2 we state the corresponding optimization problems.

3.1 Graph representation

In order to verify whether a data set actually satisfies some revealed preference axiom, it is natural to construct a graph (see Koo [19]). We now extend this procedure by taking into account a given vector $e = (e_1, \ldots, e_n)$. For some data set S, we construct the associated graph $G_e(S)$. In this graph, there is a node for every observation. Next, for each pair of observations (i, j) $(i \neq j)$, there is an arc from node i to node j when $e_i \geq p_i q_j$. The length of this arc is equal to $p_i q_j - e_i$.

The graph $G_e(S)$ will be used to test WARP, SARP and GARP. To test HARP, we make use of another graph $G'_e(S)$. The nodes and arcs of this alternative graph are defined in the same way as for the graph $G_e(S)$, but now the length of the arc is given by $\log(p_i q_j) - \log(e_i)$.

The axioms of revealed preference can then be formulated as follows:

Definition 9. ***(WARP(e))*** *The data set S satisfies WARP(e) if and only if the graph* $G_e(S)$ *does not contain any cycle consisting of two arcs.*

Definition 10. ***(SARP(e))*** *The data set S satisfies SARP(e) if and only if the graph* $G_e(S)$ *is acyclic.*

Definition 11. ***(GARP(e))*** *The data set S satisfies GARP(e) if and only if the graph $G_e(S)$ does not contain any cycles of negative length.*

Definition 12. ***(HARP(e))*** *The data set S satisfies HARP(e) if and only if the graph $G'_e(S)$ does not contain any cycles of negative length.*

3.2 Problem descriptions

We are now in a position to define an optimization problem that measures how close a given data set is to satisfying a particular axiom of revealed preference. This leads to twelve different problems. For example, for SARP(e) we obtain the problems AI-SARP, VI-SARP and HI-SARP, each corresponding to a specific index. Straightforward adaptations define the problems AI-$\{S, G, H\}$ARP, VI-$\{S, G, H\}$ARP and HI-$\{S, G, H\}$ARP. For compactness, we only state the optimization problems with respect to SARP; the optimization problems corresponding to $\{W, G, H\}$-ARP are defined analogously.

Problem 1. *(VI-SARP) Given a data set S, for what values e_i, with $0 \leq e_i \leq 1$ for each i, is $\sum_{i=1}^{n} e_i$ maximized, while S satisfies SARP(e)?*

Clearly, other objective functions are possible, We will give results and come back to this issue in Section 4.

Problem 2. *(HI-SARP) Given a data set S, what is the largest subset of observations $Q \subseteq \{1, \ldots, n\}$ such that Q satisfies SARP?*

Results concerning this problem will be given in Section 5.

Problem 3. *(AI-SARP) Given a data set S, for what value e_1, with $0 \leq e_1 \leq 1$, is e_1 maximized while S satisfies SARP(e), with $e = (e_1, \ldots, e_1)$?.*

4 The complexity of Varian's Index

Clearly, when given a vector $e = (e_1, \ldots, e_n)$, there are different ways to specify an objective function measuring the quality of e. Obvious candidates are minimize $\sum_{i=1}^{n} (1 - e_i)$, minimize $\sum_{i=1}^{n} (1 - e_i)^2$ or minimize $max_i\ (1 - e_i)$. In fact, all these objective functions can be captured by considering minimize $(\sum_{i=1}^{n} (1 - e_i)^{\rho})^{1/\rho}$ for $\rho \geq 1$. Observe that, since $\lim_{\rho \to \infty} (\sum_{i=1}^{n} (1 - e_i)^{\rho})^{1/\rho} = max_i\ (1 - e_i)$, the Afriat index arises when $\rho \to \infty$. The results in this section are phrased for $\rho = 1$, i.e., for the case where we minimize $\sum_{i=1}^{n} (1 - e_i)$ or equivalently maximize $\sum_{i=1}^{n} e_i$. At the end of the section we point out that the reduction remains valid for every fixed $\rho \geq 1$.

Let us now consider the following decision problem associated with VI-SARP:

Input: A data set $S = \{p_i, q_i \,|i = 1, \ldots, n\}$ and a number Z.
Question: Do there exist n numbers e_i, with $0 \leq e_i \leq 1$, such that

(i) The data set S satisfies SARP(e), and

(ii) $\sum_{i=1}^{n} e_i \geq Z$?

Theorem 1. *VI-SARP is NP-Hard.*

Proof. We prove that VI-SARP is NP-Hard by a reduction from the well-known NP-Hard independent set problem [16], which is formulated as follows:

Input: A graph $G = (V, E)$ and a number k.
Question: Does there exist a subset $V' \subseteq V$ of at least k vertices, such that for every pair of vertices $i, j \in V'$, the edge (i, j) is not in E?

Given an instance of IS we now construct the following instance of VI-SARP. For every node $i \in V$, there is an observation in VI-SARP: $n := |V|$. The vectors $p_i = (p_i^1, \ldots, p_i^N), q_i = (q_i^1, \ldots, q_i^N)$ are created as follows. We set, for $i = 1, \ldots, n$, $q_i^i := 1$, all remaining $q_i^j := 0$. Further, we set $p_i^i := 1$, for $i = 1, \ldots, n$. If there is an edge between node i and node j in G, i.e., if $\{i, j\} \in E$, then $p_i^j := \epsilon$ (for some $0 < \epsilon < \frac{1}{n}$), otherwise $p_i^j := 2$. Finally, we set $Z := k$. This completes the description of the instance of VI-SARP. Notice that this construction implies that if an edge exists between i and j in G, then $p_i q_j = p_j q_i = \epsilon$, else $p_i q_j = p_j q_i = 2$.

We now argue the equivalence between IS and VI-SARP. Suppose the instance of independent set is a yes-instance, i.e., an independent set of size at least k exists. For every vertex in that independent set, set $e_i = 1$ and for every other vertex set $e_i = 0$. It is clear that $\sum e_i \geq Z$. Consider the graph $G_e(S)$, and recall that an arc is present from i to j if and only if $p_i q_j \leq e_i$. We claim that the graph $G_e(S)$ is acyclic. Indeed, notice that vertices outside the independent set will not have any outgoing arcs in $G_e(S)$ since for each such vertex i: $p_i q_j - e_i = p_i q_j > 0$. Also note that no arc connects two observations corresponding to nodes in the independent set, since for a pair of such observations i, j we have $p_i q_j - e_i = p_j q_i - e_j = 2 - 1 > 0$. Thus, arcs in $G_e(S)$ only exist from vertices in the independent set to vertices outside the independent set. It follows that the graph is acyclic.

Now, suppose that the instance of VI-SARP is a yes-instance, so $\sum e_i \geq Z = k$. Then for at least k observations $e_i > \epsilon$; if not, at most $k - 1$ e_i-values exceed ϵ; since $e_i \leq 1$, $\sum e_i$ is then bounded by $k - 1 + (n - k - 1)\epsilon < k - 1 + 1 = k$, which contradicts with the requirements for a yes-instance. We will call such an e_i value *large*. We claim that the vertices with large e_i-values constitute an independent set in G. Indeed, consider two vertices i and j with a large e_i value. If i and j are connected in G, then $p_i q_j = p_j q_i = \epsilon$, implying that there is an arc in the graph $G_e(S)$ from i to j and from j to i, which is a cycle. Therefore i and j are not connected in G. Thus the set of vertices with large e_i is an independent set of size at least k. □

We now proceed with the closely related problems VI-GARP, VI-WARP and VI-HARP:

Input: A data set $S = \{p_i, q_i \,|\, i = 1, \ldots, n\}$ and a number Z.
Question: Do there exist n numbers e_i, with $0 \leq e_i \leq 1$, such that

(i) The data set S satisfies the appropriate axiom, GARP(e), WARP(e) or HARP(e), and

(ii) $\sum_{i=1}^{n} e_i \geq Z$?

Theorem 2. *VI-GARP is NP-Hard.*

Theorem 3. *VI-WARP is NP-Hard.*

Theorem 4. *VI-HARP is NP-Hard.*

Instances of these problems are built as in the proof of Theorem 1. The proofs of equivalence are relatively straightforward.

Let us now return to the general objective function $\sum_{i=1}^{n} (1 - e_i)^\rho$ (with $\rho \geq 1$) given at the start of this section. We now consider the following problem:

Input: A data set $S = \{p_i, q_i \,|i = 1, \ldots, n\}$ and a number Z.
Question: Do there exist n numbers e_i, with $0 \leq e_i \leq 1$, such that

(i) The data set S satisfies SARP(e), and

(ii) $\sum_{i=1}^{n} (1 - e_i)^{\rho} \leq Z$?

Corollary 1. *Varian's Index is NP-HARD for objective functions of the form minimize* $(\sum_{i=1}^{n} (1 - e_i)^{\rho})^{1/\rho}$*, for any fixed* $\rho \geq 1$.

Proof. Given an instance of Independent Set, create an instance of VI-$\{W, S, G, H\}$ARP as in the proof of their respective theorems with the following differences. Set $Z := n - k$ and let $0 < \epsilon < 1 - \frac{(n-k)}{(n-k+1)^{(1/\rho)}}$. It can be easily checked that the equivalences hold. □

5 The index HI

In this section, we consider the problems HI-$\{W, S, G, H\}$ARP. We give the problem HI-SARP, all other problems are analogous, differing only in the axiom of revealed preference to be satisfied. Notice that, in their original paper, Houtman and Maks already showed a relation between HI and feedback vertex set, see also [11]

Input: A data set $S = \{p_i, q_i \,|i = 1, \ldots, n\}$ and a number Z.
Question: Do there exist n numbers e_i, with $e_i \in \{0, 1\}$, such that

(i) The data set S satisfies SARP(e), and

(ii) $\sum_{i=1}^{n} e_i \geq Z$?

Theorem 5. *HI-*$\{W, S, G, H\}$*ARP is NP-Hard.*

Proof. The proof of NP-hardness for maximizing the sum of the elements of VI is easily extended to HI. As the choice of e_i is now limited to either zero or one it is clear that every large $e_i = 1$ and every other $e_j = 0$. □

Theorem 6. *No polynomial time* ρ*-approximation algorithm exists for HI-*$\{W, S, G, H\}$*ARP, unless P = NP.*

Proof. Consider an instance of Independent Set, and the corresponding instance of HI-SARP as constructed in Theorem 1. Now consider that the optimum of the HI-SARP instance is z, then the optimum for IS is also z. If not, then for we could find an independent set of size $z + 1$ and by the previous reduction we could find e so that $\sum e_i \geq z + 1$.

Now consider we have a ρ-approximation for HI-SARP, then we could find a vector-index so that $\sum e_i \geq z \times \rho$ in polynomial time. Given this vector-index we could find an independent set of size $z \times \rho$ as follows, for every i for which $e_i = 1$ add the vertex i to the independent set. This would give us a ρ-approximation for IS in polynomial time. This implies that P = NP. □

6 Afriat's index (AI)

6.1 Introductory observations

As with the previous indices, it is our goal to find the maximum value of e ($e_1 = e_2 = \ldots = e_T = e$), such that a given data set still passes $\{W, S, G, H\}$ARP.

However, such a maximum value frequently does not exist. For example, consider the following matrix of the values $p_i q_j$ (for two observations)

$$\begin{pmatrix} 1 & 0.50 \\ 0.60 & 1 \end{pmatrix}$$

As long as $e \in [0; 0.6[$, all axioms of revealed preference will be satisfied, but for $e \geq 0.6$ a cycle of negative length between the two vertices exists in both $G_e(S)$ and $G'_e(S)$ and, thus, the axioms are violated. Since there is no maximum feasible value for e, we look for the value e^* that is the supremum of the values of e for which the axioms of revealed preference are satisfied. Varian [27] describes an approximation algorithm which approximates e^* to within $(1/2)^t$ by testing the axiom under e t times. In an overview paper, Varian [29] mentions that it is also easy to calculate e^* exactly and exact values are calculated for AI-GARP in a number of papers, see for instance Choi et al. [7]. However, to the best of our knowledge, no exact polynomial algorithm has been published in the literature. In the next section we provide such a polynomial time exact algorithm for AI-$\{W, S, G\}$ARP and a separate algorithm for AI-HARP.

6.2 Complexity results

Theorem 7. *AI-WARP can be solved in $O(n^2 \log(n))$.*

Proof. We first argue that Algorithm 1 is correct. Clearly, if the dataset satisfies WARP(e), then it satisfies WARP(e') for all $e' \leq e$. Moreover, the dataset satisfies WARP(0). Thus, for an increasing e, WARP(e) becomes infeasible at some value e^*. This can only happen when an arc, completing a cycle consisting of two arcs, is added to the graph $G_e(S)$, i.e., at some value $p_i q_j$. It follows that Algorithm 1 is correct.

Next we analyse the complexity of this algorithm. To construct A, $p_i q_j$ must be calculated for all pairs of observations, which takes $O(n^2)$ time. In the worst case, this array is of size $O(n^2)$, so sorting is done in $O(n^2 \log(n))$. In the second step of the algorithm, WARP(e) is tested for different values of e. As the array is halved in each iteration, at most $O(\log(n^2))$ such tests are needed and each such test can be done in $O(n^2)$, by checking each pair of nodes for violations of WARP(e). This gives a total time complexity for the second step of $O(n^2 \log(n))$. The total time complexity is thus determined by the sorting of the array and the second step and is $O(n^2 \log(n))$. □

Algorithm 1 AI-WARP

1: Initialization: Construct an array A of all values $p_i q_j \leq 1, i \neq j$. Sort these values in ascending order.
2: Let x be the median value in A. Test WARP(x), if WARP(e) is satisfied, remove all values lower than or equal to x from A, otherwise remove all higher values.
3: If more than one element remains in the array, repeat step 2, otherwise let x be the remaining value in A, then $e^* = x$

Theorem 8. *AI-SARP can be solved in $O(n^2 \log(n))$.*

Proof. For AI-SARP we consider algorithm 1, with the adjustment that SARP(e) is tested instead of WARP(e). SARP(e) can also be tested in $O(n^2)$, for example by a topological ordering algorithm [2], leading to the same time complexity. □

Theorem 9. *AI-GARP can be solved in $O(n^{2.376}\log(n))$.*

Proof. We first note that the value e^* can be feasible for GARP(e), if for that value a cycle of length 0 exists in the graph $G(S)$. Therefore, we consider a variant of the algorithm, which does not discard the highest known feasible value of e.

The time complexity of this algorithm is similar to that for AI-WARP and AI-SARP, but differs in that testing GARP(e) takes $O(n^{2.376})$. This test is done by finding the transitive closure by way of matrix multiplication [9]. □

Finally, we provide a polynomial time algorithm for AI-HARP.

Algorithm 2 AI-HARP

1: Input: A set of observations $p_t = (p_t^1, \ldots, p_t^N), q_t = (q_t^1, \ldots, q_t^N)$ for $t = 1, \ldots, T$
2: Initialization: Construct the graph $G_1'(S)$
3: Calculate the minimum cycle mean (MCM), which is the shortest average length of the arcs in any cycle in the graph $G_1'(S)$.
4: Calculate e^* as follows: $e^* = \exp(MCM)$.

Theorem 10. *AI-HARP can be solved in time proportional to $O(n^3)$.*

Proof. We will show that computing the minimum cycle mean (MCM) of $G_1'(S)$ is sufficient to find e^*. HARP(e) is satisfied if there are no cycles of negative length in $G_e'(S)$. Thus, if such a cycle exists, we need to remove it by lowering e. A decrease in e will lengthen every arc in the graph by the same amount, as the length of an arc is $\log(p_i q_j) - \log(e)$. It is clear that the if we set the value of e^* so that the cycle with the shortest average arc length has a length of zero, the average arc length of every other cycle will be non-negative and no cycles of negative length will remain. Indeed, by setting $e^* := \exp(\text{MCM})$, the length of each arc becomes $\log(p_i q_j) - \log(\exp(\text{MCM})) = \log(p_i q_j) -$ MCM.

The time complexity of this algorithm is polynomial as there exist algorithms for finding the MCM in $O(nm)$ time [17], with m being the number of arcs in the graph. In $G_1'(S)$ there will be n^2 arcs. The building of the graph takes $O(n^2)$ time. The overall time bound of the algorithm is thus $O(n^2\log(n) + n^3) = O(n^3)$ time. □

7 Conclusion

Motivated by the increasing availability of large scale consumption data sets, we have investigated the computational complexity of testing the utility maximization hypothesis in revealed preference terms. In particular, we have focused on three goodness of fit measures for four different revealed preference axioms (i.e. WARP, SARP, GARP and HARP). We have demonstrated that, for all four axioms, the Houtman and Maks index is inapproximable and that computing Varian's index is NP-Hard. Next, we have shown that these conclusions do not apply to Afriat's index, and we have presented exact polynomial algorithms for computing this index (for every revealed preference axiom that we considered).

References

[1] S.N. Afriat. On a system of inequalities in demand analysis: an extension of the classical method. *International Economic Review*, 14(2):460–472, 1973.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.

[3] J.C.R. Alcantud, D.L. Matos, and C.R. Palmero. Goodness-of-fit in optimizing a consumer model. *Mathematical and Computer Modelling*, 52(7-8):1088–1094, 2010.

[4] J. Andreoni and J. Miller. Giving according to garp: An experimental test of the consistency of preferences for altruism. *Econometrica*, 70(2):737–753, 2002.

[5] J. Apesteguia and M.A. Ballester. A measure of rationality and welfare. Working Paper: Universitat Pompeu Fabra. Departamento de Economía y Empresa, No. 1220, 2011.

[6] D.R. Burghart, P.W. Glimcher, and S.C. Lazzaro. An expected utility maximizer walks into a bar... Working Paper, 2011.

[7] S. Choi, R. Fisman, D. Gale, and S. Kariv. Consistency and heterogeneity of individual behavior under uncertainty. *The American economic review*, 97(5):1921–1938, 2007.

[8] S. Choi, S. Kariv, W. Müller, and D. Silverman. Who is (more) rational? Technical report, National Bureau of Economic Research, 2011.

[9] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of symbolic computation*, 9(3):251–280, 1990.

[10] J.C. Cox. On testing the utility hypothesis. *The Economic Journal*, 107(443):1054–1078, 1997.

[11] M. Dean and D. Martin. How rational are your choice data? In *Conference on Revealed Preferences and Partial Identification*, 2010.

[12] F. Echenique, S. Lee, and M. Shum. The money pump as a measure of revealed preference violations. *Journal of Political Economy*, 119(6):1201–1223, 2011.

[13] P. Février and M. Visser. A study of consumer behavior using laboratory data. *Experimental economics*, 7(1):93–114, 2004.

[14] W.T. Harbaugh, K. Krause, and T.R. Berry. Garp for kids: On the development of rational choice behavior. *The American economic review*, 91(5):1539–1545, 2001.

[15] M. Houtman and J. Maks. Determining all maximal data subsets consistent with revealed preference. *Kwantitatieve methoden*, 19:89–104, 1985.

[16] R.M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 40(4):85–103, 1972.

[17] R.M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete mathematics*, 23(3):309–311, 1978.

[18] A.Y.C. Koo. An empirical test of revealed preference theory. *Econometrica*, 31(4):646–664, 1963.

[19] A.Y.C. Koo. Revealed preference–a structural analysis. *Econometrica*, 39(1):89–97, 1971.

[20] A.Y.C. Koo and G. Hasenkamp. Structure of revealed preference: some preliminary evidence. *The Journal of Political Economy*, 80(4):724–744, 1972.

[21] S.E. Landsburg. Taste change in the united kingdom, 1900-1955. *The Journal of Political Economy*, 89(1):92–104, 1981.

[22] M.E. Manser and R.J. McDonald. An analysis of substitution bias in measuring inflation, 1959-85. *Econometrica*, 56(4):909–930, 1988.

[23] A. Mattei. Full-scale real tests of consumer behavior using experimental data. *Journal of Economic Behavior & Organization*, 43(4):487–497, 2000.

[24] A. Mossin. A mean demand function and individual demand functions confronted with the weak and the strong axioms of revealed preference: an empirical test. *Econometrica*, 40(1):177–192, 1972.

[25] R. Sippel. An experiment on the pure theory of consumer's behaviour. *The Economic Journal*, 107:1431–1444, 1997.

[26] H.R. Varian. The nonparametric approach to demand analysis. *Econometrica*, 50(4):945–973, 1982.

[27] H.R. Varian. Goodness-of-fit in optimizing models. *Journal of Econometrics*, 46(1-2):125–140, 1990.

[28] H.R. Varian. Goodness-of-fit for revealed preference tests. Unpublished, 1993.

[29] H.R. Varian. Revealed preference. *Samuelsonian economics and the twenty-first century*, pages 99–116, 2006.

Laurens Cherchye
Faculty of Business and Economics - Center for Economic Studies
KU Leuven
Naamsestraat 69
8500 Kortrijk, Belgium
Email: laurens.cherchye@econ.kuleuven.be

Bram De Rock
ECARES
Université libre de Bruxelles
Av. F.D. Roosevelt 50
1050 Brussels, Belgium
Email: bderock@ulb.ac.be

Bart Smeulders
Faculty of Business and Economics - ORSTAT
KU Leuven
Naamsestraat 69
3000 Leuven, Belgium
Email: bart.smeulders@econ.kuleuven.be

Frits C.R. Spieksma
Faculty of Business and Economics - ORSTAT
KU Leuven
Naamsestraat 69
3000 Leuven, Belgium
Email: frits.spieksma@econ.kuleuven.be

Analysis and Optimization of Multi-dimensional Percentile Mechanisms

Xin Sui and Craig Boutilier and Tuomas Sandholm

Abstract

We consider the mechanism design problem for agents with single-peaked preferences over multi-dimensional domains when multiple alternatives can be chosen. Facility location and committee selection are classic embodiments of this problem. We propose a class of *percentile mechanisms*, a form of generalized median mechanisms, that are (group) strategy-proof, and derive worst-case approximation ratios for social cost and maximum load for L_1 and L_2 cost models. More importantly, we propose a sample-based framework for optimizing the choice of percentiles relative to any prior distribution over preferences, while maintaining strategy-proofness. Our empirical investigations, using social cost and maximum load as objectives, demonstrate the viability of this approach and the value of such optimized mechanisms *vis-à-vis* mechanisms derived through worst-case analysis.

1 Introduction

Mechanism design deals with design of protocols to elicit the preferences of self-interested agents so as to achieve a certain social objective. An important property of mechanisms is *strategy-proofness*, which requires that agents have no incentive to misreport their preferences to the mechanism. While payments are often used to ensure that mechanisms are strategy-proof [23, 6, 11], in many settings payments are infeasible and restrictions on preferences are required. The simple but elegant class of *single-peaked preferences* is one such example: roughly speaking, each agent has a single, most-preferred point in the alternative space and alternatives become less preferred as they are moved away from that point. In such settings, choosing a single alternative can be accomplished in a strategy-proof fashion using the famous *median mechanism* [4] and its generalizations [18, 1]. Such models are used frequently for modeling political choice, facility location, and other problems. They also have potential applications in areas such as in the design of a family of products, customer segmentation, and related tasks, as we discuss below.

Unfortunately, such mechanisms are efficient (e.g., w.r.t. social cost) only in very limited circumstances. Furthermore, allowing the choice of multiple alternatives (e.g., multiple facilities) generally causes even these limited guarantees to evaporate. In response, authors have begun to address the question of *approximate mechanism design without money* [19], which focuses on the design of strategy-proof mechanisms for problems such as multi-facility location that are approximately efficient (i.e., have good approximation ratios) [19, 15, 10]. This work provides some positive results, but is generally restricted to settings involving two facilities (alternatives) and L_2 (Euclidean) preferences.

In this paper, we propose *percentile mechanisms*—a special case of generalized median mechanisms [2, 1], but in a more general fashion. Specifically: (a) we consider selection of multiple alternatives (e.g., multi-facility location) in a multi-dimensional alternative space; (b) we address both social cost and maximum load as performance metrics; and (c) we analyze our mechanisms relative to L_1 (Manhattan) and L_2 (Euclidean) preferences. Our first contribution is the analysis of the approximation ratios of various percentile mechanisms under various assumptions. The performance guarantees of such mechanisms under worst-case assumptions are quite discouraging (much like previous results above).

Indeed, designing mechanisms that have the best possible worst-case guarantees may lead to poor performance in practice. Our second contribution is the development of a sample-based *empirical framework for the optimization of percentile mechanisms* relative to a known preference distribution. In most realistic applications of mechanism design, such as facility location, product design, and many others, the designer will have *some* knowledge of the preferences of participating agents. Assuming this takes the form of a distribution over preference profiles, we use profiles sampled from this distribution to optimize the choice of percentiles. Since the result is a percentile mechanism, strategy-proofness is maintained. Our empirical results demonstrate that, by exploiting probabilistic domain knowledge, we obtain strategy-proof mechanisms that outperform mechanisms designed to guard against worst-case profiles. Our framework can be viewed, conceptually, as a form of *automated mechanism design (AMD)*, which advocates the use of preference (or type) distributions to optimize mechanisms [7, 20].

2 Preliminaries

In this section, we introduce our model along with required concepts, notation, and motivation, and then briefly discuss a selection of related work.

2.1 The Social Choice Problem

In a standard social choice setting, we must select an *outcome* o from an outcome set O, where each of agents $i \in N = \{1, 2, \ldots, n\}$ has a preference over O. Agent preferences are represented by (weak) total order over O, or in a more precise way by a *utility function.* In our setting, we focus on the m-dimensional, *q-facility location problem* (or (m, q)-problems): we must choose q points or *locations* in an m-dimensional space $\mathbb{R}^m$ (or some bounded subspace thereof) to place facilities. Outcomes are then *location vectors* of the form $\mathbf{x} = (x_1, \ldots, x_q)$, with $x_j \in \mathbb{R}^m$ (for $j \leq q$). Each agent i has a type t_i denoting the *cost* associated with any location $x \in \mathbb{R}^m$: we write $c_i(x, t_i)$ to denote this real-valued cost. Given an outcome $\mathbf{x}$, i will use the location that has least cost, hence $c_i(\mathbf{x}, t_i) = \min_{j \leq q} c_i(x_j, t_i)$.

Facility location can be interpreted literally, and naturally models the placement of q facilities (e.g., warehouses in a supply chain, public facilities such as parks, etc.) in some geographic space. Agents will then use the least cost (or "closest") facility. However, many other choice problems fit within this class. Voting is one example [4, 1]: we can think of political candidates as being ordered along several dimensions (e.g., stance on the environment, health care, fiscal policy)—voters have preferences over points in this space—and one must elect q representatives to a committee or legislative body. In product design, a vendor may launch a family q new, related products, each described by an m-dimensional feature vector, with consumer preferences over these options leading them to select their most preferred. This also can serve as a form of customer segmentation.

In facility location problems and the other settings discussed above, it is natural to assume agent preferences are *single-peaked.* Intuitively, this means the agent has a single "ideal" location, and its cost for any chosen location increases as it "moves away from" this ideal. Formally, we don't need a distance metric, only a strict ordering on alternatives in each dimension, which is used to define a *betweenness relation.* Let $|| \cdot ||_1$ denote the L_1-norm.

Definition 1 *[2] An agent i's preference on m-dimensional space $\mathbb{R}^m$ is single-peaked if there exists a most preferred alternative $\tau(t_i)$ such that, $\forall \alpha, \beta \in \mathbb{R}^m$ satisfying $||\tau(t_i) - \beta||_1 = ||\tau(t_i) - \alpha||_1 + ||\alpha - \beta||_1$, we have $c_i(\alpha, t_i) \leq c_i(\beta, t_i)$.*

Single-peaked preferences require that if a point α lies within the "bounding box" of $\tau(t_i)$ and β, then α is at least as preferred as β. Intuitively, as we move farther away from i's ideal location $\tau(t_i)$ we can reach α via some path before we reach β. Note that this requirement does not restrict i's relative preference for α and β if neither lies within the other's bounding box (w.r.t. $\tau(t_i)$).

An agent's ideal location $\tau(t_i)$ does not fully determine its preference, even if it is single-peaked. Despite this, we will equate an agent's type t_i with its ideal location (for reasons that become clear below). However, within the class of single-peaked preferences, we can adopt specific cost functions that *are* fully determined by the ideal location t_i. Often *distance metrics* are used, and we consider both L_1 (Manhattan) and L_2 (Euclidean) distances below. Specifically, we define distance-based cost functions for i as follows:

$$c_i^p(\mathbf{x}, t_i) = \min_{j \leq q} ||t_i - x_j||_p \tag{1}$$

where $p \in \{1, 2\}$ reflects either L_1 or L_2 distance from i's nearest facility. We use $x^p[i; \mathbf{x}]$ to denote i's closest facility in the location vector $\mathbf{x}$ under the L_p-norm.

The aim in facility location is to select a set of q facilities that minimize some social objective. One natural objective is to minimize *social cost (SC)* given type profile $\mathbf{t}$, where social cost (relative to some norm p) is given by:

$$SC_p(\mathbf{x}, \mathbf{t}) = \sum_i c_i^p(\mathbf{x}, t_i) \tag{2}$$

Alternatively, we could try to balance the *load* by ensuring no facility is used by too many agents. Define the load on facility j given outcome $\mathbf{x}$ and type profile $\mathbf{t}$ as $l_j^p(\mathbf{x}, \mathbf{t}) = |\{i | x^p[i; \mathbf{x}] = j\}|$. We wish to minimize the *maximum load (ML)*, which is defined as:

$$ML_p(\mathbf{x}, \mathbf{t}) = \max_j l_j^p(\mathbf{x}, \mathbf{t}). \tag{3}$$

This objective makes sense, for instance, when a product designer launches a family of q new products, consumers purchase the product closest to their ideal product, but costs are minimized by balancing production; or when facility management costs increase superlinearly with load. Many other fundamental social objectives, such as fairness (e.g., maximum agent distance), and combinations thereof can be adopted depending on one's design goals.

2.2 Mechanisms

The goal of mechanism design is to construct mechanisms that (possibly indirectly) elicit information about agent preferences so that an outcome choice can be made that achieves some social objective. We consider *direct mechanisms* in which agents are asked to reveal their types, and an outcome is chosen based on the revealed types. In the facility location with single-peaked preferences, we consider mechanisms that ask agents to declare their ideal locations, then select an outcome $\mathbf{x}$: that is, a *mechanism* M is a function f that maps a declared type profile $\mathbf{t}$ into an outcome $f(\mathbf{t}) \in (\mathbb{R}^m)^q$ (i.e., q m-dimensional alternatives).

A mechanism f is *strategy-proof* (or truthful) if:[1]

$$c_i(f(t_i, \mathbf{t}_{-i}), t_i) \leq c_i(f(t'_i, \mathbf{t}_{-i}), t_i), \quad \forall i, t_i, t'_i, \mathbf{t}_{-i}$$

In other words, f is strategy-proof if no agent can obtain a better outcome by misreporting its true type (ideal location). *Group strategy-proofness* is defined similarly, but requires that

[1] We use *strategy-proof* to refer to dominant strategy incentive compatibility (participation is assured in our settings).

no group of agents $S \subseteq N$ can misreport their types, in a coordinated fashion, so that the outcome is better for at least one $i \in S$, and no worse for any $i \in S$.

While the ideal is to design strategy-proof mechanisms that achieve some social objective, such as minimizing social cost, this is not always feasible. In $(1, 1)$-facility location problems, if agent preferences are single-peaked, the *median mechanism*, which selects the median of all reported ideal locations, is (group) strategy-proof [4, 18] and minimizes social cost if agent preferences are all determined under a suitable distance metric (such as L_1). However, when one moves to even just two facilities, strategy-proofness and efficiency are incompatible, as demonstrated by Procaccia and Tennenholtz [19]. They propose the study of *approximate mechanisms* to handle such situations: mechanisms that are strategy-proof and come as close as possible to achieving the social objective (e.g., minimizing social cost). Formally:

Definition 2 *A mechanism f has an* approximation ratio *ε w.r.t. social objective C if:*

$$C(f(\mathbf{t}), \mathbf{t}) \leq \varepsilon \cdot \min_{\mathbf{x}} C(\mathbf{x}, \mathbf{t}).$$

We refer to such a mechanism as ε-*optimal* w.r.t. objective C (or ε-*efficient* when considering social cost/welfare). When minimizing social cost, we assume the number of agents is greater than the number of facilities (otherwise, we can trivially locate facilities at each agent's ideal to obtain a (group) strategy-proof, efficient mechanism). Notice that our mechanisms are *non-imposing*: once facilities are selected, agents are free to choose their favourite (otherwise, one can trivially minimize ML by assigning agents to facilities in an arbitrary balanced way).

2.3 Related Work

Black [4] first proposed the median mechanism for $(1, 1)$-facility location, showing it to be strategy-proof for single-peaked preferences. Moulin [18] proposed a generalized median scheme (allowing *phantom peaks*) that he proved to be the unique class of (anonymous) strategy-proof mechanisms for such preferences. Barberà *et al.* [2] later generalized this class of mechanisms further using *coalitional systems* and provided a characterization result for $(m, 1)$-problems. We refer to this class as *m-dimensional generalized median* schemes. These schemes select a location by choosing its coordinates in each dimension independently (in a "median-like" fashion).

Some work considers strategy-proof mechanisms with even more restricted preferences and domain assumptions. Border and Jordan [5] characterize strategy-proof mechanisms in m-dimensional spaces assuming *separable star-shaped* preferences (which include quadratic preferences). As in [2], location coordinates are chosen in each dimension separately. Massó and Moreno de Barreda [17] consider symmetric, single-peaked preferences (of which L_1 and L_2 are instances), and show that a mechanism is strategy-proof iff it is a *disturbed* generalized median voter schemes (which allows discontinuities). Schummer and Vohra [21] consider the problem of choosing a location on a graph (e.g., a network) relative to an extended notion of single-peakedness, obtaining positive results for trees, and negative results for cyclic graphs.

Recent attention has been focused on algorithmic aspects and approximation in strategy-proof facility location when agents have L_2 preferences. Procaccia and Tennenholtz [19] study the one-dimensional problems, and provide upper and lower bounds on the approximation ratio for social cost. Of interest here is their deterministic *left-right mechanism*, which is $(n-1)$-efficient for $(1, 2)$-problems. Lu *et al.* [15] define the (randomized) *proportional mechanism* with an approximation ratio of 4 for general distance metrics, but it cannot be applied for more than two facilities. Fotakis and Tzamos [10] show that a *winner-imposing* variant of the proportional mechanism is strategy-proof for any number of facilities, with an approximation ratio of $4q$. Escoffier *et al.* [8] define the first mechanism

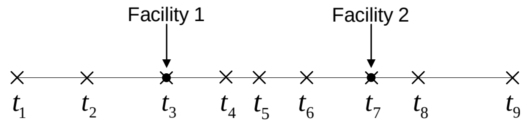


Figure 1: The $(0.25, 0.75)$-percentile mechanism for $n = 9$.

for general multi-dimensional location problems, a randomized mechanism with an approximation ratio of $n/2$, but only in the very restrictive setting where the number of agents is exactly one more than the number of facilities.

Work on load balancing games is somewhat related, but differs in that cost functions reflect the externalities agents impose on one another (by sharing a facility or some other resource). Considerable research has developed price of anarchy [12, 3] and related results. However, externalities give those models a very different character than ours.

3 Percentile Mechanisms

In this section, we introduce and analyze the class of *percentile mechanisms*, a special case of *m-dimensional generalized median mechanisms* [2, 1].

3.1 One-dimensional Percentile Mechanisms

We begin with one-dimensional facility location problems to develop intuitions. We wish to place q facilities, with each agent i having a single ideal location t_i and single-peaked preferences. Without loss of generality, we rename the agents so their ideal locations are ordered: $t_1 \leq t_2 \leq \ldots \leq t_n$. A *percentile mechanism* is specified by a vector $\mathbf{p} = (p_1, p_2, \ldots, p_q)$, where $0 \leq p_1 \leq p_2 \ldots \leq p_q \leq 1$: the $\mathbf{p}$-percentile mechanism locates the jth facility at the p_jth percentile of the reported ideal locations. In other words, the jth location is placed at $x_j = t_{i_j}$, where $i_j = \lfloor (n-1) \cdot p_j \rfloor + 1$.[2] Intuitively, we can decompose the mechanism into q independent rules, each locating one facility.

Example 1 *We illustrate the* $(0.25, 0.75)$*-percentile mechanism for a two-facility problem with* $n = 9$ *agents in Fig. 1. Ordering reported locations so that* $t_1 \leq \ldots \leq t_9$*, the mechanism locates the first facility at* $x_1 = t_3$ *(since* $\lfloor 8 \cdot 0.25 \rfloor + 1 = 3$*) and the second at* $x_2 = t_7$*.*

The following theorem shows an important property of the mechanism:

Theorem 1 *The* $\mathbf{p}$*-percentile mechanism is (group) strategyproof for any* $\mathbf{p}$*.*

Proof: We prove the theorem for the case of $q = 2$ (proofs for other cases are similar).

Let $S \subseteq N$ be a coalition of agents, $x = (x_1, x_2)$ be the location vector if agents truthfully report their ideals, and $x' = (x'_1, x'_2)$ be the location vector if agents in S jointly deviate from their peaks. In addition, let $\Delta_1 = x_1 - x'_1$ and $\Delta_2 = x'_2 - x_2$. An important observation is that, according to our mechanism, if either of Δ_1 or Δ_2 is greater or less than 0, some agent in S must be strictly worse off. We consider four cases:

I. $\Delta_1 \geq 0$ and $\Delta_2 \geq 0$. Note that we can ignore the case where both Δ_1 and Δ_2 are 0, since no agent in S gains by misreporting if neither facility moves. Assume, w.l.o.g., that $\Delta_1 > 0$ and $\Delta_2 \geq 0$. Recall that x_1 is the p_1th percentile among all reported

[2] We could equivalently use order statistics; but the percentile formulation removes dependence on the number of the agents in the mechanism's specification. It is well-known that, for any fixed n, Moulin's phantom peaks can easily be arranged to implement any order statistic.

peaks. Hence $\Delta_1 > 0$ implies that some agent $i \in S$, with $t_i \geq x_1$, reports a new ideal to the left of x_1. Agent i's cost is now:

$$c_i(x', t_i) = \min\{t_i - x'_1, x'_2 - t_i\} \geq \min\{t_i - x_1, x_2 - t_i\} = c_i(x, t_i)$$

II. $\Delta_1 \geq 0$ and $\Delta_2 < 0$. In this case, there must be an $i \in S$, with $t_i \geq x_2$, that reports a new ideal to the left of x_2; it's cost is:

$$c_i(x', t_i) = t_i - x'_2 \geq t_i - x_2 = c_i(x, t_i)$$

III. $\Delta_1 < 0$ and $\Delta_2 \geq 0$. This case is completely symmetric to Case II.

IV. $\Delta_1 < 0$ and $\Delta_2 < 0$. The case is similar to Case II: There must be an $i \in S$ whose ideal is to the right of x_2 but misreports to the left of x_2, increasing its cost.

We conclude that our percentile mechanism is (group) strategy-proof. ■

Since any percentile mechanism is strategy-proof for any class of single-peaked preferences, it prevents strategic manipulation even when applied to specific cost/preference functions. Unfortunately, percentile mechanisms can give rise to poor approximation ratios when we consider specific cost functions, specifically, L_2 or L_1 costs.[3]

Theorem 2 *Let agents have L_2 (equivalently, L_1) preferences. Let $\mathbf{p} = (p_1, p_2, \ldots, p_q)$ define a percentile mechanism M. If $q \geq 3$, the approximation ratio of M w.r.t. social cost is unbounded. The approximation ratio w.r.t maximum load is $q \cdot z$, where $z = \max_{1 \leq j \leq q}(p_{j+1} - p_{j-1})$ (defining $p_0 = 0$ and $p_{q+1} = 1$).*

The proof is provided in a longer version of the paper, but we sketch the intuitions here for the case of social cost.[4] The key point is that for any percentile vector, we can construct an ideal location profile in which the number of different peaks is exactly one more than the number of facilities, and two of the peaks are arbitrarily close. The percentile mechanism can locate one facility at each of the "close peaks," while the optimal solution will select only one of them. Since optimal social cost is arbitrarily small, an unbounded approximation ratio results.

Notice that the theorem does not hold for social cost with $q = 2$ facilities: the *left-right mechanism*, which in our terminology is the $(0, 1)$-percentile mechanism, has a bounded approximation ratio of $n - 1$ for social cost [19]. Indeed, it is not hard to show the $(0, 1)$-percentile mechanism is the *only* mechanism within the percentile family that has a bounded approximation ratio. We conjecture there is no other deterministic mechanism (even outside the percentile family) that has a bounded approximation ratio. This gives further motivation to the use of probabilistic priors to optimize the choice of percentiles (see Sec. 4).

With respect to maximum load, it is natural to ask which percentile vector $\mathbf{p}$ minimizes z in Thm. 2. We can show that the percentile mechanism that "evenly distributes" facilities is approximately optimal, and that it has the smallest approximation ratio within the family.

Proposition 1 *Let agents have L_2 (equiv. L_1) preferences. If q is odd, then the percentile mechanism with $p_j = \frac{j}{q+1}$, $\forall 1 \leq j \leq q$, is $\frac{2q}{q+1}$-optimal w.r.t. maximum load. If q is even, then the percentile mechanism with $p_j = p_{j+1} = \frac{j+1}{q+2}$, $\forall j = 2j' - 1, 1 \leq j' \leq q/2$, is $\frac{2q}{q+2}$-optimal w.r.t. maximum load.[5] In each case, the mechanism has the smallest approximation ratio within the percentile family.*

---

[3] Of course, other mechanisms, beyond simple generalized medians, depending on the preference class (e.g., disturbed median mechanisms for symmetric costs [17] like L_1 and L_2).

[4] Any omitted proofs of our main results can be found in the appendix of a longer version of this paper; see: `http://www.cs.toronto.edu/~cebly/papers.html`.

[5] For even q, the mechanism is partially imposing. We locate two facilities at each selected location, and

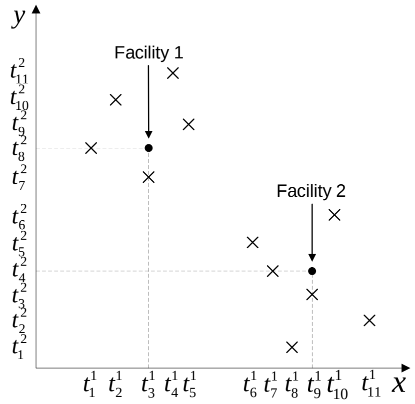


Figure 2: A percentile mechanism for a $(2, 2)$-problem when $n = 11$.

3.2 Multi-dimensional Percentile Mechanisms

As discussed above, many social choice problems can be interpreted as "facility location" problems when viewed as choice in a higher dimensional space, such as selection of political/committee representatives, product design, and the like. We now analyze a generalization of the percentile mechanism to multi-dimensional spaces.

As above, we assume that agents have single-peaked preferences (see Defn. 1). Reported types t_i are now points in $\mathbb{R}^m$. For any type profile $\mathbf{t}$, let $t_1^k \leq t_2^k \leq \ldots \leq t_n^k$ be the *ordered projection* of $\mathbf{t}$ in the kth dimension (for $k \leq m$). In other words, we simply order the reported coordinates in each dimension independently. An *m-dimensional percentile mechanism* is specified by a $q \times m$ matrix $\mathbf{P} = (\mathbf{p}_1; \mathbf{p}_2; \ldots; \mathbf{p}_q)$, where each $\mathbf{p}_j \in [0, 1]^m$ is an m-vector in the m-dimensional unit cube, with $\mathbf{p}_j = (p_j^1, p_j^2, \ldots, p_j^m)$. Given a reported profile $\mathbf{t}$, the $\mathbf{P}$-percentile mechanism locates the jth facility by selecting, for each dimension $k \leq m$, the p_j^kth percentile of the ordered projection of $\mathbf{t}$ in the kth dimension as the coordinate of facility j in that dimension. In other words:

$$x_j = (t^1_{\lfloor (n-1)\cdot p_j^1 \rfloor + 1}, t^2_{\lfloor (n-1)\cdot p_j^2 \rfloor + 1}, \ldots, t^m_{\lfloor (n-1)\cdot p_j^m \rfloor + 1}).$$

Example 2 *Fig. 2 illustrates a 2-D, two facility problem with 11 agents. With* $\mathbf{P} = (0.2, 0.7; 0.8, 0.3)$*, the* $\mathbf{P}$*-percentile mechanism locates the first facility at the x-coordinate of* t_3 *(since* $\lfloor 10 \cdot 0.2 \rfloor + 1 = 3$*) and at the y-coordinate of* t_8*; and the second facility is placed at the x-coordinate of* t_9 *and and the y-coordinate of* t_4*. Notice facilities need not be located at the ideal point of any agent.*

The following results generalize the corresponding one-dimensional results above.

Theorem 3 *The m-dimensional* $\mathbf{P}$*-percentile mechanism is (group) strategy-proof for any* $\mathbf{P}$*.*

Theorem 4 *Let agents have* L_1 *or* L_2 *preferences, and* $\mathbf{P}$ *define a percentile mechanism* M *for an* (m, q)*-facility location problem with* $m > 1$*. The approximation ratio of* M *is unbounded w.r.t. social cost for any* $\mathbf{P}$*. The approximation ratio of* M *is* $q \cdot z$ *w.r.t. maximum load, where* $z = \prod_{k=1}^m \max_{1 \leq j \leq q}(p_{j+1}^k - p_{j-1}^k)$ *(where we define* $p_0^k = 1$ *and* $p_{q+1}^k = 1$*).*

balance the agents choosing that location. Agents are indifferent to the "imposed" assignment, so this is unlike truly imposing mechanisms that remove choice from agents' hands [10]. We use this mechanism for convenience—one can define a strictly non-imposing mechanism with the same approximation ratio.

Notice that this result differs from the one-dimensional case, where the $(0, 1)$-percentile (i.e., left-right) mechanism has a bounded approximation ratio for social cost. When $m > 1$, *no* percentile mechanism has this property—this holds because the mechanism may place no facility at the ideal location of any agent. As above, however, we can optimize the percentiles for maximum load, when $q = \tilde{q}^m$ for some $\tilde{q}$ by exploiting Prop. 1 in each dimension:

Proposition 2 *Let $q = \tilde{q}^m$. If $\tilde{q}$ is odd, the mechanism that locates one facility at each percentile of the form $\frac{1}{\tilde{q}+1}$ in each dimension is $\left(\frac{2\tilde{q}}{\tilde{q}+1}\right)^m$-optimal w.r.t. maximum load. If $\tilde{q}$ is even, the mechanism that locates two facilities at each percentile of the form $\left(\frac{2}{\tilde{q}+2}\right)$ in each dimension is $\left(\frac{2\tilde{q}}{\tilde{q}+2}\right)^m$-optimal w.r.t. maximum load. Moreover, these are the smallest approximation ratios possible within the family of percentile mechanisms.*

4 Optimizing Percentile Mechanisms

We've seen that percentile mechanisms are (group) strategy-proof for general (m, q)-facility location problems, and can offer bounded approximation ratios for L_1 and L_2 preferences (though only under restricted circumstances for social cost). Unfortunately, these guarantees require optimizing the choice of percentiles w.r.t. worst-case profiles, which can sometimes lead to poor performance in practice. For example, in a $(1, 2)$-problem, decent approximation guarantees for social cost require using the $(0, 1)$-percentile mechanism; but if agent preferences are uniformly distributed in one dimension, this will, in fact, perform quite poorly. Intuitively, the $(0.25, 0.75)$-percentile mechanism should have lower expected social cost by the (probabilistically) "suitable" placement of two facilities, each for use by half of the agents.

We consider a framework for empirical optimization of percentiles within the family of percentile mechanisms that should admit much better performance in practice. As in automated mechanism design [7, 20], we assume a prior distribution D over agent preference profiles. Hence agent preferences can be correlated in our model. One will often assume a prior model D (e.g., learned from observation) that renders individual agent preferences independent *given* that model, but this is not required. In many practical settings, such as facility location or product design, such distributional information will in fact be readily available. We sample preference profiles from this distribution, and use them to optimize the percentiles in the $\mathbf{P}$ matrix to ensure the best possible expected performance w.r.t. our social objective.

Unlike classic AMD, we restrict ourselves to the specific family of percentile mechanisms. While this limits the space of mechanisms, we do this for several reasons. First, it provides a much more compact mechanism parameterization over which to optimize than in typical AMD settings.[6] Second, since the resulting mechanism is (group) strategy-proof no matter which percentiles are chosen, the optimization need not account for incentive constraints (unlike standard AMD). Of course, when considering specific classes of single-peaked preferences, such as L_1 or L_2 costs as we do here, a wider class of strategy-proof mechanisms could be used (e.g., disturbed median mechanisms [17]); but these have more parameters, and as we will see below empirically, they are unlikely to offer any better performance—since our optimized percentile mechanisms achieve near-optimal social cost. In addition, errors due to sampling, or even misestimation of the prior D, have no impact on the strategyproofness of the mechanism. Third, unlike Bayesian optimization—in other words, methods that choose optimal facility placement relative to the prior with *no elicitation* of ideal locations—optimized percentile mechanisms are *responsive* to the specific preferences of the agents.

[6] AMD has been explored in a parameterized mechanism space, e.g., in combinatorial auctions [13, 14].

Distribution		$q=2$	$q=3$	$q=4$
D_u	SC	$(0.25, 0.75)$	$(0.16, 0.5, 0.84)$	$(0.12, 0.37, 0.63, 0.88)$
	ML	$(0.49, 0.50)$	$(0.33, 0.35, 0.98)$	$(0.25, 0.26, 0.74, 0.75)$
D_g	SC	$(0.25, 0.75)$	$(0.15, 0.5, 0.85)$	$(0.1, 0.35, 0.65, 0.9)$
	ML	$(0.49, 0.50)$	$(0.33, 0.35, 0.9)$	$(0.25, 0.26, 0.74, 0.75)$
D_{gm}	SC	$(0.17, 0.68)$	$(0.16, 0.59, 0.93)$	$(0.12, 0.37, 0.68, 0.94)$
	ML	$(0.49, 0.50)$	$(0.14, 0.65, 0.66)$	$(0.17, 0.34, 0.73, 0.74)$

Table 1: Optimal percentiles for different distributions, objectives, and numbers of facilities.

Let agent type profiles $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ be drawn from distribution D. Given a $\mathbf{P}$-percentile mechanism, let $f_{\mathbf{P}}(\mathbf{t})$ denote the chosen locations when the agent type profile is $\mathbf{t}$. The goal is to select $\mathbf{P}$ to minimize the expected social cost or maximum load:

$$\min_{\mathbf{P}} \mathbb{E}_D \left[SC_p(f_{\mathbf{P}}(\mathbf{t}), \mathbf{t})\right]; \text{ or } \min_{\mathbf{P}} \mathbb{E}_D \left[ML_p(f_{\mathbf{P}}(\mathbf{t}), \mathbf{t})\right]$$

Naturally, other objectives can be modelled in this way too.

Given Y sampled preference profiles, we optimize percentile selection relative to the Y sampled profiles. In our experiments below, we use simple numerical optimization for this purpose. Specifically, we consider all possible values for the percentile matrix $\mathbf{P}$. For each of them, we compute the average social cost (maximum load) over Y sample profiles, and select the one that has the minimum objective value. Alternatively, one can formulate the minimization problem as a mixed integer programming (MIP) for L_1 costs, or a mixed integer quadratically constained program (MIQCP) for L_2 costs, and use standard optimization tools, e.g., CPLEX, to solve the problem. However, determining concise formulations is non-trivial and effective use of these formulations is left to future research.[7]

In the following experiments, we consider problems with $n = 101$ agents, with agent preferences drawn independently from three classes of distributions: uniform D_u, Gaussian D_g and mixture of Gaussians D_{gm} with 2 or 3 components. Each distribution reflects rather different assumptions about agent preferences: that they are spread evenly (D_u); that they are biased toward one specific location (D_g); or that they partitioned into 2 or 3 loose clusters (D_{gm}). In all cases, $T = 500$ sampled profiles are used for optimization. We examine results for both social cost and maximum load.

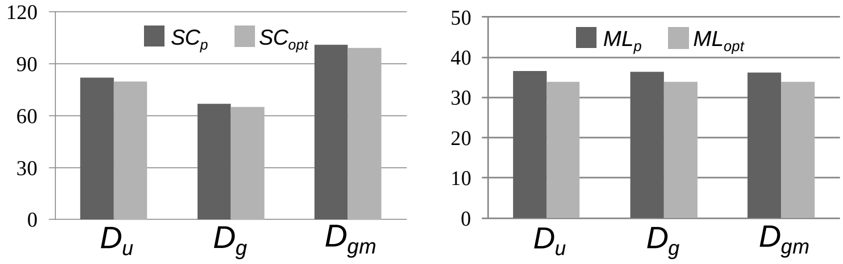
One-dimensional mechanisms

We begin with simple one-dimensional problems with $q = 2$, 3 or 4. Table 1 shows the percentiles resulting from our optimization for both SC and ML under each of the three distributions.[8] For example, when agent ideal locations are uniformly distributed, the $(0.25, 0.75)$-percentile mechanism minimizes the expected social cost for two facilities. This is expected, since the uniform (and Gaussian) distribution partitions agents into two groups of roughly equal size, and facilities should be located at the median positions of each group.

The performance of the optimized percentile mechanisms is extremely good. Fig. 3 compares the expected social cost and maximum load of our mechanisms with those given by *optimal placement* of facilities (results for $q = 3$ are shown, but others are similar). Recognize however that optimal placement is not realizable with any strategy-proof mechanism. Despite this, optimized percentile mechanisms perform nearly as well in expectation in all three cases. Contrast this with the performance of the mechanisms with provable approximation ratios. When $q = 2$, the $(0, 1)$-percentile mechanism has an average social cost

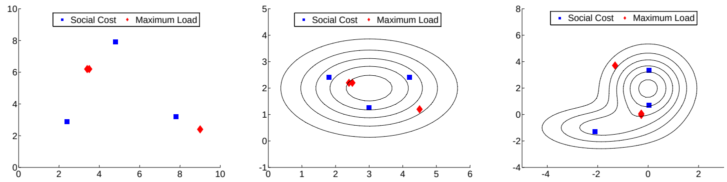
[7] We describe preliminary formulations of the MIP and MIQCP, which do not scale well, in the appendix of a longer version of this paper; see: `http://www.cs.toronto.edu/~cebly/papers.html`.

[8] D_u is uniform on $[0, 10]$. D_g is Gaussian $\mathcal{N}(0, 2)$ with $\mu = 0, \sigma^2 = 2$. D_{gm} is a Gaussian mixture with 3 components: $\mathcal{N}(-4, 4)$ (weight 0.4), $\mathcal{N}(0, 1)$ (weight 0.45), and $\mathcal{N}(5, 2)$ (weight 0.15).



(a) Social Cost (b) Maximum Load

Figure 3: Comparison of optimized percentile mechanism and optimal value ($q = 3$).



Distribution		Percentile/OPT
D_u	SC	871.2/832.4
	ML	51/51
D_g	SC	424.8/417.38
	ML	51/51
D_{gm}	SC	696.5/681.46
	ML	51/51

Figure 4: Optimized Percentiles for (a) **2D**: Uniform, (b) **2D**: Gaussian, (c) **2D**: Gaussian mixture, and (d) **4D**.

of 242.4, 340.9 and 523.2 for D_u, D_g and D_{gm}, respectively; but the social cost of our mechanisms are only 123.7, 76.5, and 165.1, respectively. When $q = 3$, the $(0.25, 0.5, 0.75)$-percentile mechanism has the best approximation ratio for ML (see Prop. 1). Its average maximum loads are 39.5, 38.7 and 38.3, which are close to (but not as good as) the loads of the optimized percentile mechanisms (36.5, 36.5, and 36.2).

Multi-dimensional mechanisms

We also experimented with two additional problems. **2D** is a $(2, 3)$-problem where agents have L_2 preferences, capturing, say, the placement of three public projects like libraries, or warehouses. **4D** is a $(4, 2)$-problem with L_1 preferences, which might model the selection of 2 products for launch, each with four attributes that predict consumer demand.[9]

For the problem **2D** we show the *expected placement* of facilities given the selected percentiles in Fig. 4(a)-(c), for both SC and ML, for each of the three distributions. (*Actual* facility placement will shift to match the reported type profile in each instance.) Placement for SC tends to be distributed appropriately, while ML places two facilities adjacent to one another. For **4D**, we measure performances rather than visualizing locations. Fig. 4(d) compares expected SC and ML of our optimized percentile mechanisms to those using true optimal facility placements: the percentile mechanisms are always optimal for ML;[10] and for SC, non-strategy-proof optimal placements are only 1.75%-4.45% better than placements using our optimized, strategy-proof mechanisms.[11] This strongly suggests that percentile mechanisms, optimized using priors over preferences, are well-suited to multi-dimensional, single-peaked domains.

[9]For **2D**, D_u is uniform over $[0, 10]$ in each dimension. D_g is normal with mean $\boldsymbol{\mu} = [3, 2]$ and covariance $\boldsymbol{\Sigma} = [2, 1]\mathbf{I}$. D_{gm} is a 2 component mixture: $\mathcal{N}([-2, -1], [2, 1]\mathbf{I})$ (weight 0.3) and $\mathcal{N}([0, 2], [1, 3]\mathbf{I})$ (weight 0.7). For **4D**, D_u is uniform over $[0, 10]$ in each dimension. D_g is $\mathcal{N}([3, 2, 1, 2], [2, 3, 4, 1]\mathbf{I})$. D_{gm} is a 2 component mixture: $\mathcal{N}([2, 1, 0, 1], [4, 6, 8, 5]\mathbf{I})$ (weight 0.4) and $\mathcal{N}([1, 2, 1, 0], [7, 4, 5, 8]\mathbf{I})$ (weight 0.6).

[10]This comes from the fact that the mechanism always locates two facilities at almost the same position, and achieves optimal maximum load. However, this is not always possible for more than two facilities.

[11]Computing the optimal solution in the multi-dimensional problem is NP-hard, so we use K-means clustering algorithms as approximations.

5 Conclusion and Future Research

We proposed a family of *percentile mechanisms* for multi-dimensional, multi-facility location problems, designed to be (group) strategy-proof when preferences are single-peaked. Using different costs measures, we derived several approximation ratios. We also developed a sample-based framework for optimizing percentile mechanisms that, much like automated mechanism design, exploits priors over preferences. Our empirical results demonstrate the power of this approach, showing social objectives can be optimized much more effectively than is possible using mechanisms with tight worst-case performance guarantees (indeed, our mechanisms provide close to optimal results in practice).

This work is a starting point for the design of optimized mechanisms for single-peaked domains, and can be extended in a number of ways. Obviously one can consider mechanisms for other classes of (single-peaked) preferences (e.g., quadratic [5] or symmetric single-peaked [17]). Other social objectives should be explored, including those that combine various desiderata (such as SC and ML), and those that trade off facility cost with benefit (e.g., additional facilities decrease social cost, but the expense must be factored in as well [16]). Additional development of the optimization models needed for percentile mechanisms (e.g., our MIP or MIQCP formulations) are needed to make our approach more practical; preliminary experiments suggest that local search techniques may be very promising in this respect. Sample complexity results are also of interest. Finally, incremental (or multi-stage) mechanisms that trade off social cost, communication costs, and agent privacy [9, 22] would be extremely valuable in our setting.

References

[1] Salvador Barberà. Strategy-proof social choice. In K. J. Arrow, A. K. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 2, pages 731–832. North-Holland, Amsterdam, 2010.

[2] Salvador Barberà, Faruk Gul, and Ennio Stacchetti. Generalized median voter schemes and committees. *Journal of Economic Theory*, 61(2):262–289, 1993.

[3] Vöcking Berthold. Selfish load balancing. In Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*, pages 517–542. Cambridge University Press, 2007.

[4] Duncan Black. On the rationale of group decision-making. *Journal of Political Economy*, 56(1):23–34, 1948.

[5] Kim C. Border and J. S. Jordan. Straightforward elections, unanimity and phantom voters. *The Review of Economic Studies*, 50(1):153–170, 1983.

[6] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.

[7] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 103–110, Edmonton, 2002.

[8] Bruno Escoffier, Laurent Gourvès, Thang Nguyen Kim, Fanny Pascual, and Olivier Spanjaard. Strategy-proof mechanisms for facility location games with many facilities. In *Proceedings of the Second International Conference on Algorithmic Decision Theory (ADT-11)*, pages 67–81, Piscataway, NJ, 2011.

[9] Joan Feigenbaum, Aaron D. Jaggard, and Michael Schapira. Approximate privacy: Foundations and quantification (extended abstract). In *Proceedings of the Eleventh*

ACM Conference on Electronic Commerce (EC'10), pages 167–178, Cambridge, MA, 2010.

[10] Dimitris Fotakis and Christos Tzamos. Winner-imposing strategyproof mechanisms for multiple facility location games. In *Proceedings of the Sixth International Workshop on Internet and Network Economics (WINE-10)*, pages 234–245, Stanford, CA, 2010.

[11] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[12] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science (STACS-99)*, pages 404–413, Trier, DE, 1999.

[13] Anton Likhodedov and Tuomas Sandholm. Methods for boosting revenue in combinatorial auctions. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 232–237, Pittsburgh, PA, 2004.

[14] Anton Likhodedov and Tuomas Sandholm. Approximating revenue-maximizing combinatorial auctions. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*, pages 267–274, Pittsburgh, PA, 2005.

[15] Pinyan Lu, Xiaorui Sun, Yajun Wang, and Zeyuan A. Zhu. Asymptotically optimal strategy-proof mechanisms for two-facility games. In *Proceedings of the Eleventh ACM Conference on Electronic Commerce (EC'10)*, pages 315–324, Cambridge, MA, 2010.

[16] Tyler Lu and Craig Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the Twenty-second International Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 280–286, Barcelona, Spain, 2011.

[17] Jordi Massó and Inés Moreno de Barreda. On strategy-proofness and symmetric single-peakedness. *Games and Economic Behavior*, 72(2):467–484, 2011.

[18] Hervé Moulin. On strategy-proofness and single peakedness. *Public Choice*, 35(4):437–455, 1980.

[19] Ariel D. Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. In *Proceedings of the Tenth ACM Conference on Electronic Commerce (EC'09)*, pages 177–186, Stanford, CA, 2009.

[20] Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP-03)*, Kinsale, Ireland, 2003.

[21] James Schummer and Rakesh V. Vohra. Strategy-proof location on a network. *Journal of Economic Theory*, 104(2):405–428, 2002.

[22] Xin Sui and Craig Boutilier. Efficiency and privacy tradeoffs in mechanism design. In *Proceedings of the Twenty-fifth AAAI Conference on Artificial Intelligence (AAAI-11)*, pages 738–744, San Francisco, CA, 2011.

[23] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

Xin Sui
Dept. of Computer Science
University of Toronto
Toronto, ON, CANADA
Email: `xsui@cs.toronto.edu`

Craig Boutilier
Dept. of Computer Science
University of Toronto
Toronto, ON, CANADA
Email: `cebly@cs.toronto.edu`

Tuomas Sandholm
Computer Science Dept.
Carnegie Mellon University
Pittsburgh, PA, USA
Email: `sandholm@cs.cmu.edu`

Empirical Aspects of Plurality Election Equilibria

David R. M. Thompson, Omer Lev, Kevin Leyton-Brown
and Jeffrey S. Rosenschein

Abstract

Social choice functions aggregate the different preferences of agents, choosing from a set of alternatives. Most research on manipulation of voting methods studies (1) limited solution concepts, (2) limited preferences, or (3) scenarios with a few manipulators that have a common goal. In contrast, we study voting in plurality elections through the lens of Nash equilibrium, which allows for the possibility that any number of agents, with arbitrary different goals, could all be manipulators. We do this through a computational analysis, leveraging recent advances in (Bayes-)Nash equilibrium computation for large games. Although plurality has exponentially many pure-strategy Nash equilibria, we demonstrate how a simple equilibrium refinement—assuming that agents very weakly prefer to vote truthfully—dramatically reduces this set. We also use symmetric Bayes-Nash equilibria to investigate the case where voters are uncertain of each others' preferences. Although our refinement does not completely eliminate the problem of multiple equilibria, it tends to predict an increased probability that a good candidate will be selected (e.g., the candidate that would win if voters were truthful, or a Condorcet winner).

1 Introduction

When multiple agents have differing preferences, voting mechanisms are often used to decide among the alternatives. One desirable property for a voting mechanism is strategy-proofness, i.e., that it is optimal for agents to truthfully report their preferences. However, the Gibbard-Satterthwaite theorem [12; 27] shows that no non-dictatorial strategy-proof mechanism can exist. Whatever other desirable properties a voting mechanism may have, there will always be the possibility that some participant can gain by voting strategically.

Since voters may vote strategically (i.e., manipulate or counter-manipulate) to influence an election's results, according to their knowledge or perceptions of others' preferences, much research has considered ways of limiting manipulation. This can be done by exploiting the computability limits of manipulations (e.g., finding voting mechanisms for which computing a beneficial manipulation is NP-hard [2; 1; 30]), by limiting the range of preferences (e.g., if preferences are single-peaked, there exist non-manipulable mechanisms [10]), randomization [13; 25], etc.

When studying the problem of vote manipulation, nearly all research falls into two categories: coalitional manipulation and equilibrium analysis. Much research into coalitional manipulation considers models in which a group of truthful voters faces a group of manipulators who share a common goal. Less attention has been given to Nash equilibrium analysis which models the (arguably more realistic) situation where all voters are potential manipulators. One reason is that it is difficult to make crisp statements about this problem: strategic voting scenarios give rise to a multitude of Nash equilibria, many of which involve implausible outcomes. For example, even a candidate who is ranked last by all voters can be unanimously elected in a Nash equilibrium—observe that when facing this strategy profile, no voter gains from changing his vote.

Despite these difficulties, this paper considers the Nash (and subsequently, Bayes-Nash) equilibria of voting games. We focus on plurality, as it is by far the most common voting mechanism used in practice. We refine the set of equilibria by adding a small additional assumption: that agents realize a very small gain in utility from voting truthfully; we call this restriction a *truthfulness incentive*. We ensure that this incentive is small enough that it is always overwhelmed by the opportunity to be pivotal between any two candidates: that is, a voter always has a greater preference for swinging an election in the direction of his preference than for voting truthfully. All the same, this restriction is powerful enough to rule out the bad equilibrium described above, as well as being, in our view, a good model of reality, as voters often express a preference for voting truthfully.

Dutta and Laslier [7] studied a somewhat similar model, where voters have a lexigraphic preference for truthfulness. They demonstrated that for some voting mechanism, a small preference for truthfulness can eliminate all pure-strategy Nash equilibria. We observed a similar occurrence in our results with plurality (which is problematic voting methods designed to reach an equilibrium by an iterative process, e.g., [21; 19]).

We take a computational approach to the problem of characterizing the Nash equilibria of voting games. This has not previously been done in the literature, because the resulting normal-form games are enormous. For example, representing our games (10 players and 5 candidates) in the normal form would require about a hundred million payoffs. Unsurprisingly, these games are intractable for current equilibrium-finding algorithms, which have worst-case runtimes exponential in the size of their inputs. We overcame this obstacle by leveraging recent advances in compact game representations and efficient algorithms for computing equilibria of such games, specifically action-graph games [15; 14] and the support-enumeration method [28].

Our first contribution is an equilibrium analysis of full-information models of plurality elections. We analyze how many Nash equilibria exist when truthfulness incentives are present. We also examine the winners, asking questions like how often they also win the election in which all voters vote truthfully, or how often they are also Condorcet winners. We also investigate the social welfare of equilibria; for example, we find that it is very uncommon for the worst-case result to occur in equilibrium.

Our second contribution involves the possibly more realistic scenario in which the information available to voters is incomplete. We assume that voters know only a probability distribution over the preference orders of others, and hence identify Bayes-Nash equilibria. We found that although the truthfulness incentive eliminates the most implausible equilibria (i.e., where the vote is unanimous and completely independent of the voters preferences), many other equilibria remain. Similarly to Duverger's law (which claims that plurality election systems favor a two-party result [9], but does not directly apply to our setting), we found that a close race between almost any pair of candidates was possible in equilibrium. Equilibria supporting three or more candidates were possible, but less common.

1.1 Related Work

Analyzing equilibria in voting scenarios has been the subject of much work, with many researchers proposing various frameworks with limits and presumptions to deal with both the sheer number of equilibria, and to deal with more real-life situations, where there is limited information. Early work in this area, by McKelvey and Wendell [20], allowed for abstention, and defined an equilibrium as one with a Condorcet winner. As this is a very strong requirement, such an equilibrium does not always exist, but they established some criteria for this equilibrium that depends on voters' utilities.

Myerson and Weber [23] wrote an influential article dealing with the Nash equilibria of voting games. Their model assumes that players only know the probability of a tie occurring between each pair of players, and that players may abstain (for which they have a slight preference). They show that multiple equilibria exist, and note problems with Nash equilibrium as a solution concept in this setting. The model was further studied and expanded in subsequent research [4; 16]. Assuming a slightly different model, Messner and Polborn [22], dealing with perturbations (i.e., the possibility that the recorded vote will be different than intended), showed that equilibria only includes two candidates ("Duverger's law"). Our results, using a different model of partial information (Bayes-Nash), show that with the truthfulness incentive, there is a certain predilection towards such equilibria, but it is far from universal.

Looking at iterative processes makes handling the complexity of considering all players as manipulators simpler. Dhillon and Lockwood [6] dealt with the large number of equilibria by using an iterative process that eliminates weakly dominated strategies (a requirement also in Feddersen and Pesendorfer's definition of equilibrium [11]), and showed criteria for an election to result in a single winner via this process. Using a different process, Meir et al. [21] and Lev and Rosenschein [19] used an iterative process to reach a Nash equilibrium, allowing players to change their strategies after an initial vote with the aim of myopically maximizing utility at each stage.

Dealing more specifically with the case of abstentions, Desmedt and Elkind [5] examined both a Nash equilibrium (with complete information of others' preferences) and an iterative voting protocol, in which every voter is aware of the behavior of previous voters (a model somewhat similar to that considered by Xia and Contizer [29]). Their model assumes that voting has a positive cost, which encourages voters to abstain; this is similar in spirit to our model's incentive for voting truthfully, although in this case voters are driven to withdraw from the mechanism rather than to participate. However, their results in the simultaneous vote are sensitive to their specific model's properties.

Rewarding truthfulness with a small utility has been used in some research, though not in our settings. Laslier and Weibull [18] encouraged truthfulness by inserting a small amount of randomness to jury-type games, resulting in a unique truthful equilibrium. Dutta and Laslier [7] attempted to inject truthfulness directly into a voting rule combined of approval voting and veto, but only found a few existence results that show truthful equilibria exist in that case. A more general result has been shown in Dutta and Sen [8], where they included a subset of participants which, as in our model, would vote truthfully if it would not change the result. They show that in such cases, many social choice functions (those that satisfy the No Veto Power) are Nash-implementable, i.e., there exists a mechanism in which Nash equilibria correspond to the voting rule. However, as they acknowledge, the mechanism is highly synthetic, and, in general, implementability does not help us understand voting and elections, as we have a predetermined mechanism.

2 Definitions

Before detailing our specific scenario, we first define elections, and how winners are determined.

Elections are made up of candidates, voters, and a mechanism to decide upon a winner:

Definition 1. *Let C be a set of m candidates, and let A be the set of all possible preference orders over C. Let V be a set of n voters, and every voter $v_i \in V$ has some element in A which is his true, "real" value (which we shall mark as a_i), and some element of A which he announces as his value, which we shall denote as $\tilde{a}_i$.*

Note that our definition of a voter incorporates the possibility of him announcing a value different than his true value (strategic voting).

Definition 2. *A voting rule is a function $f : A^n \to 2^C \setminus \emptyset$.*

In this paper, we restrict our attention to plurality, where a point is given to each voter's most-preferred candidate, and the candidates with the highest score win.

Our definition of voting rules allows for multiple winners. However, in many cases what is desired is a single winner; in these cases, a tie-breaking rule is required.

Definition 3. *A tie-breaking rule is a function* $t : 2^C \to C$ *that, given a set of elements in* C*, chooses one of them as a (unique) winner.*

There can be many types of tie-breaking rules, such as random or deterministic, lexical or arbitrary. In this work, we use a lexical tie-breaking rule.

Another important concept is that of a Condorcet winner.

Definition 4. *A Condorcet winner is a candidate* $c \in C$ *such that for every other candidate* $d \in C$ $(d \neq c)$ *the number of voters that rank* c *over* d *is at least* $\lceil \frac{n}{2} \rceil$*.*

Condorcet winners do not exist in every voting scenario, and many voting rules—including plurality—are not Condorcet-consistent (i.e., even when there is a Condorcet winner, that candidate may lose). Note that our definition allows for the possibility of multiple Condorcet winners in a single election, in cases where n is even. Conversely, a Condorcet loser is ranked below any other candidate by a majority of voters.

To reason about the equilibria of voting systems, we need to formally describe them as games, and hence to map agents' preference relations to utility functions. More formally, each agent i must have a utility function $u_i : A^n \mapsto \mathbb{R}$, where $u_i(a_V) > u_i(a'_V)$ indicates that i prefers the outcome where all the agents have voted a_V over the outcome where the agents vote a'_V. Representing preferences as utilities rather than explicit rankings allows for the case where i is uncertain what outcome will occur. This can arise either because he is uncertain about the outcome given the agents' actions (because of random tie-breaking rules), or because he is uncertain about the actions the other agents will take (either because they are behaving randomly, or because they have committed to a strategy that agent i does not observe). In this paper, we assume that an agent's utility only depends on the candidate that gets elected and on his own actions (e.g., an agent can strictly prefer to abstain when his vote is not pivotal, as in [5], or to vote truthfully). Thus, we obtain simpler utility functions $u_i : C \times A \mapsto \mathbb{R}$, with an agent i's preference for outcome a_V denoted $u_i(t(f(a_V)), \tilde{a}_i)$.

In this paper, we consider two models of games, full-information games and symmetric Bayesian games. In both models, each agent must choose an action $\tilde{a}_i$ without conditioning on any information revealed by the voting method or by the other agents. In a full-information game, each agent has a fixed utility function which is common knowledge to all the others. In a symmetric Bayesian game, each agent's utility function (or "type") is an i.i.d. draw from a commonly known distribution of the space of possible utility functions, and each agent must choose an action without knowing the types of the other agents, while seeking to maximize his expected utility.

We consider a plurality voting setting with 10 voters and 5 candidates (numbers chosen to give a setting both computable and with a range of candidates), and with the voters' preferences chosen randomly. Suppose voter i has a preference order of $a^5 \succ a^4 \succ \ldots \succ a^1$, and the winner when voters voted a_V is a^j. We then define i's utility function as

$$u_i(f(t(a_V)), \tilde{a}_i) = u_i(a^j, \tilde{a}_i) = \begin{cases} j & a_i \neq \tilde{a}_i \\ j + \epsilon & a_i = \tilde{a}_i, \end{cases}$$

with $\epsilon = 10^{-6}$.

In the incomplete-information case, we model agents as having one of six possible types (to make the problem more easily computable), each corresponding to a different (randomly selected) preference ordering. The agent's type draws are i.i.d. but the probability of each type is not necessarily uniform. Instead, the probability of each type is drawn from a uniform distribution, and then normalized; thus, the probabilities ranged from 0.0002 to 0.55.

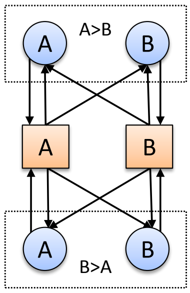


Figure 1: An action graph game encoding of a simple two-candidate plurality vote. Each round node represents an action that a voter can choose. Dashed-line boxes define which actions are open to a voter given his preferences; in a Bayesian AGG, an agent's type determines the box from which he is allowed to choose his actions. Each square node is an adder, tallying the number of votes a candidate received.

3 Method

Before we can use any Nash-equilibrium-finding algorithm, we need to represent our games in a form that the algorithm can use. Because normal form games require space exponential in the number of players, they are not practical for games with more than a few players. The literature contains many "compact" game representations that require exponentially less space to store games of interest, such as congestion [26], graphical [17], and action-graph games [15]. Action-graph games (AGGs) are the most useful for our purposes, because they are very compactly expressive (i.e., if the other representations can encode a game in polynomial-space then AGGs can as well), and fast tools have been implemented for working with them.

Action-graph games achieve compactness by exploiting two kinds of structure in a game's payoffs: anonymity and context-specific independence. Anonymity means that an agent's payoff depends only on his own action and the number of agents who played each action. Context-specific independence means that an agent's payoff depends only on a simple sufficient statistic that summarizes the joint actions of the other players. Both properties apply to our games: plurality treats voters anonymously, and selects candidates based on simple ballot counts.

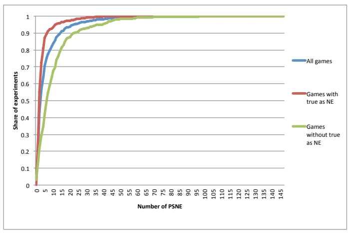
Encoding our voting games as action-graph games is relatively straightforward. For each set of voters with identical preferences, we create one action node for each possible way of voting. For each candidate, we create an adder node that counts how many votes the candidate receives. Directed edges encode which vote actions contribute to a candidate's score, and that every action's payoff can depend on the scores of all the candidates (see Figure 1).

A variety of Nash-equilibrium-finding algorithms exist for action-graph games [15; 3]. In this work, we used the support enumeration method [24; 28] exclusively because it allows Nash equilibrium enumeration. This algorithm works by iterating over possible supports, testing each for the existence of a Nash equilibrium. In the worst case, this requires exponential time, but in practice SEM's heuristics (exploiting symmetry and conditional dominance) enable it to find all the pure-strategy Nash equilibria of a game quickly.

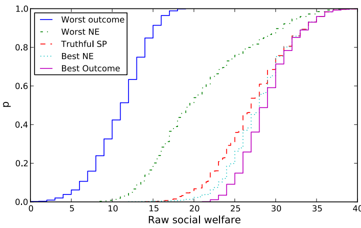
We represented our symmetric Bayesian games using a Bayesian game extension to action-graph games [14]. Because we were concerned only with *symmetric* pure Bayes-Nash equilibria, it remained feasible to search for every equilibrium with SEM.

4 Pure-Strategy Nash Equilibrium Results

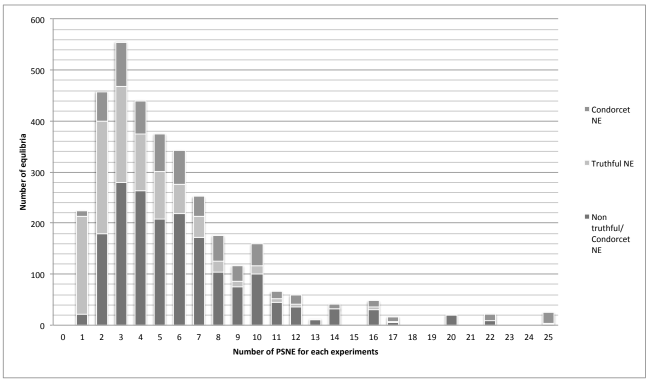
To examine pure strategies, we ran $1,000$ voting experiments using plurality with 10 voters and 5 candidates. Such a game might ordinarily have hundreds of thousands of Nash equilibria. However, adding a small truthfulness incentive ($\epsilon = 10^{-6}$) lowers these numbers significantly. Not counting permutations of voters with the same preferences, every game had 25 or fewer equilibria; counting permutations, the maximum number of equilibria was still only 146. Indeed, an overwhelming number of these games (96.2%) had fewer than 10 equilibria (27 with permutations). More surprisingly, a few (1.1%) had no pure Nash equilibria at all.[1] To gauge the impact of the truthfulness incentive, we also ran 50 experiments without it; every one of these games had over a hundred thousand equilibria, without even considering permutations.



(a) CDF showing the fraction of games having the given number of pure-strategy Nash equilibria or fewer (including permutations)



(b) The social welfare distribution



(c) The number of truthful and Condorcet winning equilibria, depending on total number of equilibria per experiment. Note that in the "tail", the data is based on only a few experiments.

Figure 2: Equilibria and social welfare in Plurality

We shall examine two aspects of the results: the preponderance of equilibria with victors being the voting method's winners,[2] and Condorcet winners. Then, moving to the wider concept of social welfare of the equilibria (possible due to the existence of utility functions), we examine both the social welfare of the truthful voting rule vs. best and worse possible Nash equilibria, as well as the average rank of the winners in the various equilibria.

[1]This is especially relevant to voting procedures relying on the existence of pure Nash equilibrium, and seeking to "find" one, such as the one proposed in [21].

[2]This, when expanded to more voting rules, may be an interesting comparative criterion between voting mechanisms.

For 63.3% of the games, the truthful preferences were a Nash equilibrium, but more interestingly, many of the Nash equilibria reached, in fact, the same result as the truthful preferences: 80.4% of the games had at least one equilibrium with the truthful result, and looking at the multitudes of equilibria, the average share of truthful equilibrium (i.e., result was the same as with truthful vote) was 41.56% (out of games with a truthful result as an equilibrium, the share was 51.69%). Without the truthfulness incentive, the average share of truthful equilibrium was 21.77%.

Looking at Condorcet winners, 92.3% of games had Condorcet winners, but they were truthful winners only in 44.7% of the games (not a surprising result, as plurality is far from being Condorcet consistent). However, out of all the equilibria, the average share of equilibria with a victorious Condorcet winner was 40.14% (of games which had a Condorcet winner the average share is 43.49%; when the Condorcet winner was also the truthful winner, its average share of equilibria is 56.96%).

Looking at the wider picture (see Figure 2c), the addition of the truthful incentive made possible games with very few Nash equilibria. They, very often, resulted in the truthful winner. As the number of equilibria grows, the truthful winner part becomes smaller, as the Condorcet winner part increases.

Turning to look at the social welfare of equilibria, once again, the existence of the truthfulness incentive enables us to reach "better" equilibria. In 92.8% of the cases, the worst-case outcome was not possible at all (recall that without the truthfulness incentive, every result is possible in some Nash equilibrium), while only in 29.7% of cases, the best outcome was not possible. We note that while truthful voting led to the best possible outcome in 59% of cases, it is still stochastically dominated by best-case Nash equilibrium (see Figure 2b).

When looking at the distribution of welfare throughout the multitudes of equilibria, one can see that the concentration of the equilibria is around high-ranking candidates, as the average share of equilibria by candidates with an average ranking (across all voters in the election) of less than 1 was 56.38%. Even if we exclude Condorcet winners (as they, on many occasions, are highly ranked), the average ranking of less than 1 was 46.56% (excluding truthful winners resulted in 27.48% with average ranking less than 1). Fully 71.65%, on average, of the winners in every experiment had above (or equal) the median rank, and in more than half the experiments (52.3%) all equilibria winners had a larger score than the median. As a comparison, the numbers from experiments without the truthfulness incentive, are quite different: candidates—whatever their average rank—won, with minor fluctuations, about the same number of equilibria (57% of winners, were, on average, above or equal to the median rank).

5 Bayes-Nash Equilibria Results

Moving beyond the full-information assumption, we considered plurality votes where the agents have incomplete information about each other's preferences. In particular, we assumed that the agents have i.i.d. (but not necessarily uniformly distributed) preferences, and that each agent knows only his own preferences and the commonly-known prior distribution. Again, we considered the case of 10 voters and 5 candidates, but now also introduced 6 possible types for each voter. For each of 50 games, we computed the set of all symmetric pure-strategy Bayes-Nash equilibria, both with and without the ϵ-truthfulness incentive.

Our first concern was studying how many equilibria each game had and how the truthfulness incentive affected the number of equilibria. The set of equilibria was small (< 28 in every game) when the truthfulness incentive was present. Surprisingly, only a few equilibria were added when the incentive was relaxed. In fact, in the majority of games (76%), there

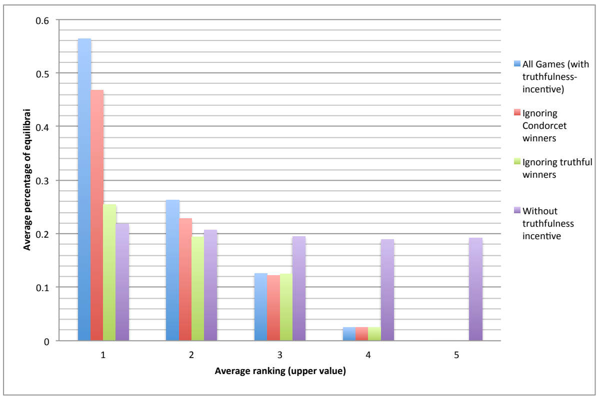


Figure 3: The average proportion of equilibria won by candidates with average rank of 0–1, 1–2, etc.

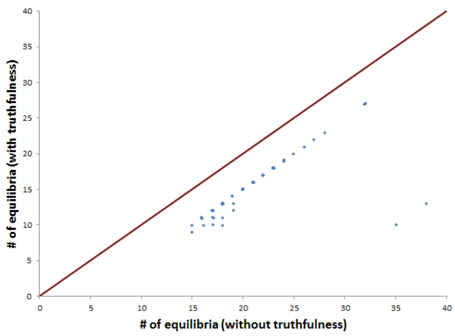


Figure 4: The number of symmetric pure-strategy Bayes-Nash equilibria in plurality votes with and without the ϵ-truthfulness incentive

were exactly five new equilibria: one for each strategy profile where all types vote for a single candidate (see Figure 4).

Looking into the structure of these equilibria, we found two interesting, and seemingly contradictory, properties: most equilibria (95.2%) only involved two or three candidates (i.e., voters only voted for a limited set of candidates), but every candidate was involved in some equilibrium. Thus, we can identify an equilibrium by the number of candidates it involves (see Figure 5). Notably, most equilibria involved only two candidates, with each type voting for their most preferred candidate of the pair. Further, most games had 10 such equilibria, one for every possible pair. There were two reasons why some pairs of candidates did not have corresponding equilibria in some games. First, sometimes one candidate Pareto-dominated the other (i.e., was preferred by every type). Second, sometimes the types that liked one candidate were so unlikely to be sampled that close races were extremely low probability (relative to ϵ); in such cases, agents preferred to be deterministically truthful than pivotal with very small probability.[3] This observation allowed us to derive a theoretical

[3]There were two outlier games where one of the types had a very low probability (< 0.001). Because of this, the probability of a realization where half the agents had this type approached machine-ϵ. Thus, any pure strategy profile where this type votes one way and all the other types vote another way will result in a 2-candidate equilibrium (20 such 2-candidate combinations exist, so these games had 20 additional two-candidate equilibria.)

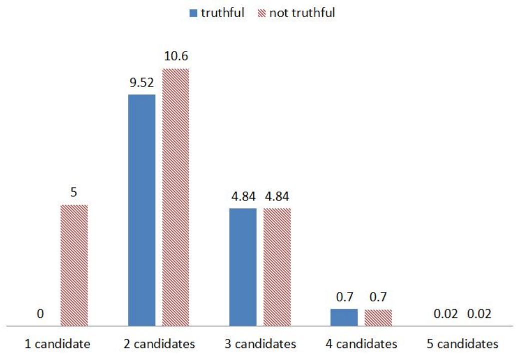


Figure 5: Every instance had many equilibria, most of which only involved a few candidates.

result about when a 2-candidate equilibrium will exist.

Let ℓ be the minimal difference between the utility of 2 different candidates, across all voters (in our scenarios, this minimal difference is 1).

Proposition 5. *In a plurality election with a truthfulness incentive of* ϵ*, as long as* $(\frac{1}{n})^{\lfloor \frac{n}{2} \rfloor}\ell \geq \epsilon$*, for every* $c_1, c_2 \in C$*, either* c_1 *Pareto dominates* c_2 *(i.e., all voters rank* c_1 *higher than* c_2*), or there exists a pure Bayes-Nash equilibrium in which each voter votes for his most preferred among these two candidates.*

Due to space constraints, we provide only proof sketch.

Proof sketch. Let us define a strategy as follows: every voter that prefers c_1 over c_2 votes for c_1; otherwise, he votes for c_2. Obviously, if c_2 is Pareto dominated, every individual voter believes that he will be better off voting truthfully, and this may not be an equilibrium. However, if c_2 is not Pareto dominated, then there is a probability larger than (or equal to) $\frac{1}{n}$ that there is a voter who prefers c_2 to c_1. Hence, the probability that a voter who prefers c_1 to c_2 will be pivotal is at least $(\frac{1}{n})^{\lfloor \frac{n}{2} \rfloor}$. If the benefit to all voters from being pivotal in this way is larger than ϵ, the value of the truthfulness incentive, the voter will not deviate from that strategy. Thus, when $(\frac{1}{n})^{\lfloor \frac{n}{2} \rfloor}\ell \geq \epsilon$ they do not deviate. □

These two-candidate equilibria have some interesting properties. Because they can include any two candidates that do not Pareto-dominate each other, it is possible for them to exclude a third candidate that Pareto-dominates both. In this way, it is possible for two-candidate equilibria to fail to elect a Condorcet winner. However, because every two-candidate equilibrium is effectively a pairwise runoff, it is impossible for a two-candidate equilibrium to elect a Condorcet loser.

Equilibria supporting three or more candidates are less straightforward. Which 3-candidate combinations are possible in equilibrium (even without ϵ-truthful incentives) can depend on the specific type distribution and the agents' particular utilities. Also, in these equilibria, agents do not always vote for their most preferred of the three alternatives (again, depending on relative probabilities and utilities). Finally, 3-candidate equilibria can elect a Condorcet loser with non-zero probability.

6 Discussion and Future Work

Our work approaches the issues of voting manipulation by combining two less-common approaches: assuming all voters are manipulators, rather than just a subset with a shared goal, and looking at Nash equilibria as a whole, rather than searching for other solution concepts or a specific equilibrium. We utilized only a small and realistic assumption—that users attach a small value to voting their truthful preferences. Using the AGG framework to analyze the Nash equilibria and symmetric Bayes-Nash equilibria of plurality, we can extrapolate from the data and reveal properties of such voting games.

We saw several interesting results, beyond a reduction in the number of equilibria, due to our truthfulness incentive. One of the most significant was the "clustering" of many equilibria around candidates that can be viewed as resembling the voters' intention. A very large share of each game's equilibria resulted in winners that were either truthful winners (according to plurality) or Condorcet winners. Truthful winners were selected in a larger fraction of equilibria when the total number of equilibria was fairly small (as was the case in a large majority of our experiments), and their share decreased as the number of equilibria increased (where we saw, in cases where there were Condorcet winners, that those equilibria took a fairly large share of the total).

Looking at social welfare enabled us to compare equilibrium outcomes to all other possible outcomes. We observed that plurality achieved nearly the best social welfare possible (a result that did not rely on our truthfulness incentive). While another metric showed the same "clustering" we noted above, most equilibrium results concentrated around candidates that were ranked, on average, very high (on average, more than 50% of winners in every experiment had a rank less than 1). This, in a sense, raises the issue of the rationale of seeking to minimize the amount of manipulation, as we found that manipulation by all voters very often results in socially beneficial results.

In the Bayes-Nash results, we saw that lack of information generally pushed equilibria to be a "battle" between a subset of the candidates—usually two candidates (as Duverger's law would indicate), but occasionally more.

There is much more work to be done in the vein we have introduced in this paper. This includes examining the effects of varying the number of voters and candidates, changing utility functions, as well as looking at more voting rules and determining properties of their equilibria. Voting rules can be ranked according to their level of clustering, how good, socially, their truthful results are, and other similar criteria. Furthermore, it would be worthwhile to examine other distributions of preferences and preference rules, such as single-peaked preferences. Computational tools can also be useful to assess the usefulness of various strategies available to candidates (e.g., it might be more productive for a candidate to attack a weak candidate to alter the distribution).

7 Acknowledgments

Authors would like to thank the iAGT semester in the Hebrew University, where this work was initiated, and would also like to thank Reshef Meir for helpful discussions on the topic of this work. This work was partially supported by Israel Ministry of Science and Technology grant #3-6797, and the Google Inter-University Center for Electronic Markets and Auctions.

References

[1] John J. Bartholdi III and James B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8:341–354, 1991.

[2] John J. Bartholdi III, Craig A. Tovey, and Michael A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[3] Constantinos Daskalakis, Grant Schoenebeck, Gregory Valiant, and Paul Valiant. On the complexity of Nash equilibria of action-graph games. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009.

[4] Francesco De Sinopoli. On the generic finiteness of equilibrium outcomes in plurality games. *Games and Economic Behavior*, 34(2):270–286, February 2001.

[5] Yvo Desmedt and Edith Elkind. Equilibria of plurality voting with abstentions. In *Proceedings of the 11th ACM conference on Electronic commerce (EC '10)*, pages 347–356, Cambridge, Massachusetts, USA, June 2010.

[6] Amrita Dhillon and Ben Lockwood. When are plurality rule voting games dominance-solvable? *Games and Economic Behavior*, 46(1):55–75, January 2004.

[7] Bhaskar Dutta and Jean-François Laslier. Costless honesty in voting. in 10th International Meeting of the Society for Social Choice and Welfare, Moscow, 2010.

[8] Bhaskar Dutta and Arunava Sen. Nash implementation with partially honest individuals. *Games and Economic Behavior*, 74(1):154–169, January 2012.

[9] Maurice Duverger. *Political Parties: Their Organization and Activity in the Modern State*. Methuen Publishing, 1959.

[10] Bruno Escoffier, Jérôme Lang, and Meltem Öztürk. Single-peaked consistency and its complexity. In *Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 366–370, Amsterdam, The Netherlands, 2008. IOS Press.

[11] Timothy Feddersen and Wolfgang Pesendorfer. Voting behavior and information aggregation in elections with private information. *Econometrica*, 65(5):1029–1058, September 1997.

[12] Allan Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–602, July 1973.

[13] Allan Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665–681, April 1977.

[14] Albert Xin Jiang and Kevin Leyton-Brown. Bayesian Action-Graph Games. In *Proceedings of the 24th Conference on Neural Information Processing Systems (NIPS)*, 2010.

[15] Albert Xin Jiang, Kevin Leyton-Brown, and Navin A.R. Bhat. Action-graph games. *Games and Economic Behavior*, 71:141–173, 2011.

[16] Kei Kawai and Yasutora Watanabe. Inferring strategic voting. Technical report, Northwestern, 2010.

[17] M.J. Kearns, M.L. Littman, and S.P. Singh. Graphical models for game theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI)*, 2001.

[18] Jean-François Laslier and Jörgen W. Weibull. A strategy-proof condorcet jury theorem. forthcoming, Scandinavian Journal of Economics, 2012.

[19] Omer Lev and Jeffrey S. Rosenschein. Convergence of iterative voting. In *Proceedings of the 11th International Coference on Autonomous Agents and Multiagent Systems (AAMAS)*, Valencia, Spain, June 2012.

[20] Richard D. McKelvey and Richard E. Wendell. Voting equilibria in multidimensional choice spaces. *Mathematics of Operations Research*, 1(2):144–158, May 1976.

[21] Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, and Nicholas R. Jennings. Convergence to equilibria of plurality voting. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*, pages 823–828, Atlanta, July 2010.

[22] Matthias Messner and Mattias K. Polborn. Miscounts, duverger's law and duverger's hypothesis. Technical Report 380, IGIER (Innocenzo Gasparini Institute for Economic Research), Bocconi University, 2011.

[23] Roger B. Myerson and Robert J. Weber. A theory of voting equilibria. *The American Political Science Review*, 87(1):102–114, March 1993.

[24] R. W. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63:642–662, 2009.

[25] Ariel D. Procaccia. Can approximation circumvent Gibbard-Satterthwaite? In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*, pages 836–841, Atlanta, Georgia, USA, July 2010.

[26] R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *Int. J. Game Theory*, 2:65–67, 1973.

[27] Mark Allen Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, April 1975.

[28] David R. M. Thompson, Samantha Leung, and Kevin Leyton-Brown. Computing nash equilibria of action-graph games via support enumeration. In *Proceedings of the 7th international conference on Internet and Network Economics (WINE)*, 2011.

[29] Lirong Xia and Vincent Conitzer. Stackelberg voting games: Computational aspects and paradoxes. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*, pages 805–810, Atlanta, Georgia, USA, 2010.

[30] Lirong Xia, Michael Zuckerman, Ariel D. Procaccia, Vincent Conitzer, and Jeffrey S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, Pasadena, California, USA, July 2009.

David R. M. Thompson
Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada, V6T 1Z4
Email: `daveth@cs.ubc.edu`

Omer Lev
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Jerusalem 91904, Israel
Email: `omerl@cs.huji.ac.il`

Kevin Leyton-Brown
Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada, V6T 1Z4
Email: `kevinlb@cs.ubc.ca`

Jeffery S. Rosenchein
School of Computer Science and Engineering
The Hebrew University of Jerusalem
Jerusalem 91904, Israel
Email: `jeff@cs.huji.ac.il`

How Many Vote Operations Are Needed to Manipulate A Voting System?

Lirong Xia

Abstract

In this paper, we propose a framework to study a general class of strategic behavior in voting, which we call *vote operations*. Our main theorem is the following: if we fix the number of alternatives, generate n votes i.i.d. according to a distribution π, and let n go to infinity, then, for any $\epsilon > 0$, with probability at least $1 - \epsilon$, the minimum number of operations that are necessary for the strategic individual to achieve her goal falls into one of the following four categories: (1) 0, (2) $\Theta(\sqrt{n})$, (3) $\Theta(n)$, and (4) ∞. This theorem holds for any set of vote operations, any individual vote distribution π, and any *integer generalized scoring rule*, which includes (but is not limited to) most commonly studied voting rules, e.g., approval voting, all positional scoring rules (including Borda, plurality, and veto), plurality with runoff, Bucklin, Copeland, maximin, STV, and ranked pairs.

We also show that many well-studied types of strategic behavior fall under our framework, including (but not limited to) constructive/destructive manipulation, bribery, and control by adding/deleting votes, margin of victory, and minimum manipulation coalition size. Therefore, our main theorem naturally applies to these problems.

1 Introduction

Voting is a popular method used to aggregate voters' preferences to make a joint decision. One of the most desired properties for voting rules is *strategy-proofness*, that is, no voter has incentive to misreport her preferences to obtain a better outcome of the election. Unfortunately, strategy-proofness is not compatible with some other natural desired properties, due to the celebrated Gibbard-Satterthwaite theorem [14, 22], which states that when there are at least three alternatives, no strategy-proof voting rule satisfies the following two natural properties: non-imposition (every alternative can win) and non-dictatorship (no voter is a dictator whose top ranked alternative is always the winner).

Even though manipulation is inevitable, researchers have set out to investigate whether computational complexity can serve as a barrier against various types of strategic behavior, including manipulation. The idea is, if we can prove that it is computationally too costly for a strategic individual to find a beneficial operation, she may give up doing so. Initiated by Bartholdi, Tovey, and Trick [2], a fair amount of work has been done in the computational social choice community to characterize the computational complexity of various types of strategic behavior, including the following. See [10, 12, 21] for recent surveys.

• *Manipulation*: a voter or a coalition of voters cast false vote(s) to change the winner (and the new winner is more preferred).

• *Bribery*: a strategic individual changes some votes by bribing the voters to make the winner preferable to her [9]. The bribery problem is closely related to the problem of computing the *margin of victory* [5, 16, 31].

• *Control*: a strategic individual adds or deletes votes to make the winner more preferable to her [3]. Control by adding votes is equivalent to *false-name manipulation* [6].

Most previous results in "using computational complexity as a barrier against strategic behavior" are worst-case analyses of computational complexity. Recently, an increasing number of results show that manipulation, as a particular type of strategic behavior, is typically not hard to compute. One direction, mainly pursued in the theoretical computer science community, is to obtain a quantitative version of the Gibbard-Satterthwaite theorem, showing that for any given voting rule that is "far" enough from any dictatorships, an instance of manipulation can be found easily with high probability. This line of research was initiated by Friedgut, Kalai, and Nisan [13], where they proved the theorem for 3 alternatives and neutral voting rules. The theorem was extended to an arbitrary number of alternatives by Isaksson, Kindler, and Mossel [15], and finally, the neutrality constraint was removed by Mossel and Racz [17]. Other extensions include Dobzinski and Procaccia [8] and Xia and Conitzer [33].

Another line of research is to characterize the "frequency of manipulability", defined as the probability for a randomly generated preference-profile to be manipulable by a group of manipulators, where the non-manipulators' votes are generated i.i.d. according to some distribution (for example, the uniform distribution over all possible types of preferences). Peleg [18], Baharad and Neeman [1], and Slinko [23, 24] studied the asymptotic value of the frequency of manipulability for positional scoring rules when the non-manipulators' votes are drawn i.i.d. uniformly at random. Procaccia and Rosenschein [20] showed that for positional scoring rules, when the non-manipulators votes are drawn i.i.d. according to any distribution that satisfies some natural conditions, if the number of manipulators is $o(\sqrt{n})$, where n is the number of non-manipulators, then the probability that the manipulators can succeed goes to 0 as n goes to infinity; if the number of manipulator is $\omega(n)$, then the probability that the manipulators can succeed goes to 1.

This dichotomy theorem was generalized to a class of voting rules called *generalized scoring rules (GSRs)* by Xia and Conitzer [32]. A GSR is defined by two functions f, g, where f maps each vote to a vector in multidimensional space, called a *generalized scoring vector* (the dimensionality of the space is not necessarily the same as the number of alternatives). Given a profile P, let *total generalized scoring vector* be the sum of $f(V)$ for each vote V in P. Then, g selects the winner based on the total preorder of the components of the total generalized scoring vector. We call a GSR a *integer* GSR, if the components of all generalized scoring vectors are integers. (Integer) GSRs are a general class of voting rules. One evidence is: many commonly studied voting rules are integer GSRs, including (but not limited to) approval voting, all positional scoring rules (which include Borda, plurality, and veto), plurality with runoff, Bucklin, Copeland, maximin, STV, and ranked pairs. [1] As another evidence, GSRs admit a natural axiomatic characterization [34].

While most of the aforementioned results are about manipulation, in this paper, we focus the optimization variants of various types of strategic behavior, including manipulation, bribery, and control. Despite being natural, to the best of our knowledge, such optimization variants have been investigated for only three types of strategic behavior. The first is the *unweighted coalitional optimization (UCO)* problem, where we are asked to compute the minimum number of manipulators who can make a given alternative win [37]. Approximation algorithms have been proposed for UCO for specific voting systems, including positional scoring rules and maximin [35–37]. The second is the *margin of victory* problem, where we are asked to compute the smallest number of voters who can change their votes to change the winner [5, 16, 31]. The third is the *minimum manipulation coalition size* problem, which is similar to the margin of victory, except that all voters who change their votes must prefer the new winner to the old winner [19].

1.1 Our Contributions

In this paper, we introduce a unified framework to study a class of strategic behavior for generalized scoring rules, which we call *vote operations*. In our framework, a strategic individual seeks to change the winner by applying some operations, which are modeled as vectors in a multidimensional space. We study three goals of the strategic individual: (1) making a favored alternative win, called *constructive vote operation (*CVO), (2) making a disfavored alternative lose, called *destructive vote operation (*DVO*)*, and (3) change the winner of the election, called *change-winner vote operation*

[1] The definition of these commonly studied voting rules can be found in, e.g., [32]. In this paper, we define GSRs as voting rules where the inputs are profiles of linear orders. GSRs can be easily generalized to include other types of voting rules where the inputs are not necessarily linear orders, for example, approval voting.

*(*CWVO*).* The framework will be formally defined in Section 3. This is our main conceptual contribution.

Our main technical contribution is the following asymptotical characterization of the number of operations that are necessary for the strategic individual to achieve her goal.

Theorem 1 (informally put) Fix the number of alternatives and the set of vote operations. For any integer generalized scoring rule and any distribution π over votes, we generate n votes i.i.d. according to π and let n go to infinity. Then, for any VO $\in \{\text{CVO}, \text{DVO}, \text{CWVO}\}$ and any $\epsilon > 0$, with probability at least $1 - \epsilon$, the minimum number of operations that are necessary for the strategic individual to achieve VO is one of the following: (1) 0, (2) $\Theta(\sqrt{n})$, (3) $\Theta(n)$, and (4) ∞.

More informally, Theorem 1 states that in large elections, to achieve a specific goal (one of the three goals described above), with probability that can be infinitely close to 1 the strategic individual needs to either do nothing (the goal is already achieved), apply $\Theta(\sqrt{n})$ vote operations, apply $\Theta(n)$ vote operations, or the goal cannot be achieve no matter how many vote operations are applied. This characterization holds for any integer generalized scoring rule, any set of vote operations, and any distribution π for individual votes.

The proof of Theorem 1 is based on the Central Limit Theorem and on sensitivity analyses for the integer linear programmings (ILPs). It works as follows. We will formulate each of the strategic individual's three goals as a set of ILPs in Section 4. By applying Central Limit Theorem, we show that with probability that goes to 1 the random generated preference-profile satisfies a desired property. Then, for each such preference-profile we apply the sensitivity analyses in [7] to show that with high probability the number of operations that are necessary is either 0, $\Theta(\sqrt{n})$, $\Theta(n)$, or ∞.

While Theorem 1 looks quite abstract, we show later in the paper that many well-studied types of strategic behavior fall under our vote operation framework, including constructive/destructive manipulation, bribery, and control by adding/deleting votes, margin of victory, and minimum manipulation coalition size.[2] Therefore, we naturally obtain corollaries of Theorem 1 for these types of strategic behavior. Of course our theorem applies to other types of strategic behavior, for example the mixture of any types mentioned above, which is known as *multimode control attacks* [11].

1.2 Related Work and Discussion

To the best of our knowledge, we are the first to do the following in the voting setting: (1) study manipulation, bribery, and control under a unified framework and (2) in this unified framework, model the strategic individual's goals as ILPs and conduct sensitivity analyses. Our main theorem applies to any integer generalized scoring rule for destructive manipulation, constructive and destructive bribery and control by adding/deleting votes, where no similar results were obtained even for specific voting rules. Three previous papers obtained similar results for specific types of strategic behavior. The applications of our main theorem to these types of strategic behavior are slightly weaker, but we stress that our main theorem is significantly more general.

Three related papers. First, the dichotomy theorem in [32] implies that, (informally) when the votes are drawn i.i.d. from *some* distribution, with probability that goes to 1 the solution to constructive and destructive UCO is either 0 or approximately $\sqrt{n}$ for *some* favored alternatives. However, this result only works for the UCO problem and some distributions over the votes.

Second, it was proved in [31] that for any non-redundant generalized scoring rules that satisfy a continuity condition, when the votes are drawn i.i.d. and we let the number of voters n go to infinity, either with probability that can be arbitrarily close to 1 the margin of victory is $\Theta(\sqrt{n})$, or with probability that can be arbitrarily close to 1 the margin of victory is $\Theta(n)$. It is easy to show that for non-redundant voting rules, the margin of victory is never 0 or ∞. Though it was shown in [31] that many commonly studied voting rules are GSRs that satisfy such continuity condition, in general it

[2]We defer the definition of these types of strategic behavior to Section 6.

is not clear how restrictive the continuity condition is. More importantly, the result only works for the margin of victory problem.

Third, in [19], the authors investigated the distribution over the minimum manipulation coalition size for positional scoring rules when the votes are drawn i.i.d. from the uniform distribution. However, it is not clear how their techniques can be extended beyond the uniform distributions and positional scoring rules, which are a very special case of generalized scoring rules. Moreover, the paper only focused on the minimum manipulation coalition size problem.

Our results has both negative and positive implications. On the negative side, our results provide yet another evidence that computational complexity is not a strong barrier against strategic behavior, because the strategic individual now has some information about the number of operations that are needed, without spending any computational cost or even without looking at the input instance. Although the estimation of our theorem may not be very precise (because we do not know which of the four cases a given instance belongs to), such estimation may be explored to designing effective algorithms that facilitate strategic behavior. On the positive side, this easiness of computation is not always a bad thing: sometimes we want to do such computation in order to test how robust a given preference-profile is. For example, computing the margin of victory is an important component in designing novel *risk-limiting audit methods* [5, 16, 25–29, 31].

While being quite general, our results have two main limitations. First, they are asymptotical results, where we fix the number of alternatives and let the number of voters go to infinity. We do not know the convergence rate, or equivalently, how many voters are needed for the observation to hold. In fact, this is a standard setting in previous work, especially in the studies of "frequency of manipulability". We feel that our results work well in settings where there are small number of alternatives and large number of voters, e.g., political elections. Second, our results show that with high probability one of the four cases holds (0, $\Theta(\sqrt{n})$, $\Theta(n)$, ∞), but we do not know which case holds more often. It is possible to refine our study for specific voting rules and specific types of strategic behavior that fall under our framework, which we leave as future work.

2 Preliminaries

Let $\mathcal{C}$ denote the set of *alternatives*, $|\mathcal{C}| = m$. We assume strict preference orders. That is, a vote is a linear order over $\mathcal{C}$. The set of all linear orders over $\mathcal{C}$ is denoted by $L(\mathcal{C})$. A *preference-profile* P is a collection of n votes for some $n \in \mathbb{N}$, that is, $P \in L(\mathcal{C})^n$. Let $L(\mathcal{C})^* = \bigcup_{n=1}^{\infty} L(\mathcal{C})^n$. A *voting rule* r is a mapping that assigns to each preference-profile a single winner. That is, $r : L(\mathcal{C})^* \to \mathcal{C}$. Throughout the paper, we let n denote the number of votes and m denote the number of alternatives.

We now recall the definition of *generalized scoring rules (GSRs)* [32]. For any $K \in \mathbb{N}$, let $\mathcal{O}_K = \{o_1, \ldots, o_K\}$. A *total preorder* (*preorder* for short) is a reflexive, transitive, and total relation. Let $\text{Pre}(\mathcal{O}_k)$ denote the set of all preorders over $\mathcal{O}_K$. For any $\vec{p} \in \mathbb{R}^K$, we let $\text{Ord}(\vec{p})$ denote the preorder $\trianglerighteq$ over $\mathcal{O}_K$ where $o_{k_1} \trianglerighteq o_{k_2}$ if and only if $p_{k_1} \geq p_{k_2}$. That is, the k_1-th component of $\vec{p}$ is as large as the k_2-th component of $\vec{p}$. For any preorder $\trianglerighteq$, if $o \trianglerighteq o'$ and $o' \trianglerighteq o$, then we write $o =_{\trianglerighteq} o'$. Each preorder $\trianglerighteq$ naturally induces a (partial) strict order $\triangleright$, where $o \triangleright o'$ if and only if $o \trianglerighteq o'$ and $o' \ntrianglerighteq o$.

Definition 1 *Let $K \in \mathbb{N}$, $f : L(\mathcal{C}) \to \mathbb{R}^K$ and $g : Pre(\mathcal{O}_K) \to \mathcal{C}$. f and g determine a* generalized scoring rule (GSR) *$GS(f, g)$ as follows. For any preference-profile $P = (V_1, \ldots, V_n) \in L(\mathcal{C})^n$, abusing the notation we let $f(P) = \sum_{i=1}^{n} f(V_i)$, and let $GS(f, g)(P) = g(Ord(f(P)))$. We say that $GS(f, g)$ is* of order K.

When for all $V \in L(\mathcal{C})$, $f(V) \in \mathbb{Z}^K$, we call $GS(f, g)$ an integer GSR.

For any $V \in L(\mathcal{C})$, $f(V)$ is called a *generalized scoring vector*, $f(P)$ is called a *total generalized scoring vector*, and $\text{Ord}(f(P))$ is called the *induced preorder* of P. The class of integer GSRs is equivalent to the class of *rational* GSRs, where the components of each generalized scoring vector is in $\mathbb{Q}$, because for any $l > 0$, $\text{GS}(f, g) = \text{GS}(l \cdot f, g)$.

Most commonly studied voting rules are generalized scoring rules, including (but not limited to) approval voting, Bucklin, Copeland, maximin, plurality with runoff, ranked pairs, and multi-stage voting rules that use GSRs in each stage to eliminate alternatives (including Nanson's and Baldwin's rule). As an example, we recall the proof from [32] that the *single transferable vote (STV)* rule (a.k.a. *instant-runoff voting* or *alternative vote* for single-winner elections) is an integer generalized scoring rule.

Example 1 *STV selects the winner in m rounds. In each round, the alternative that gets the lowest plurality score (the number of times that the alternative is ranked in the top position) drops out, and is removed from all of the votes (so that votes for this alternative transfer to another alternative in the next round). Ties are broken alphabetically. The last-remaining alternative is the winner.*

To see that STV is an integer GSR, we will use generalized scoring vectors with many components. For every proper subset S of alternatives, for every alternative c outside of S, there is a component in the vector that contains the number of times that c is ranked first if all of the alternatives in S are removed. Let
- $K_{STV} = \sum_{i=0}^{m-1} \binom{m}{i}(m-i)$*; the components are indexed by (S, j), where S is a proper subset of $\mathcal{C}$ and $j \leq m, c_j \notin S$.*
- $(f_{STV}(V))_{(S,j)} = 1$*, if after removing S from V, c_j is at the top; otherwise* $(f_{STV}(V))_{(S,j)} = 0$.
- g_{STV} *selects the winners based on $\trianglerighteq$ as follows. In the first round, let j_1 be the index such that $o_{(\emptyset,j_1)}$ is ranked the lowest in $\trianglerighteq$ among all $o_{(\emptyset,j)}$ (if there are multiple such j's, then we break ties alphabetically to select the least-preferred one). Let $S_1 = \{c_{j_1}\}$. Then, for any $2 \leq i \leq m-1$, define S_i recursively as follows: $S_i = S_{i-1} \cup \{j_i\}$, where j_i is the index such that $o_{(S_{i-1},j_i)}$ is ranked the lowest in $\trianglerighteq$ among all $o_{(S_{i-1},j)}$; finally, the winner is the unique alternative in $(\mathcal{C} \setminus S_{m-1})$.*

Another evidence on the generality of GSRs is that GSRs admit a natural axiomatic characterization [34]. That is, GSRs are the class of voting rules that satisfy *anonymity, homogeneity,* and *finite local consistency*. Anonymity says that the winner does not depend on the name of the voters, homogeneity says that if we duplicate the preference-profile multiple times, the winner does not change, and finite local consistency is an approximation to the well-studied *consistency* axiom. Not all voting rules are GSRs, for example, *Dodgson's rule* is not a GSR because it does not satisfy homogeneity [4], and the following skewed majority rule is also not a GSR because it also violates homogeneity.

Example 2 *For any $0 < \gamma < 1$, the* γ-majority *rule is defined for two alternatives $\{a, b\}$ as follows: b is the winner if and only if the number of voters who prefer b is more than the number of voters who prefer a by at least n^γ.*

Admittedly, these γ-majority rules are quite artificial. Later in this paper we will see that the observation made for GSRs in our main theorem (Theorem 1) does not hold for γ-majority rules for any $\gamma \neq 1/2$.

3 The Unified Framework

All types of strategic behavior mentioned in the introduction have the following characteristics in common. The strategic individual (who can be a group of manipulators, a briber, or a controller, etc.) changes the winner by changing the votes in the preference-profile. Therefore, for generalized scoring rules, any such an operation is uniquely represented by changes in the total generalized scoring vector. This is in contrast to some other types of strategic behavior where the strategic individual changes the set of alternatives or the voting rule [3, 30]. In this section, we first define the set of operations the strategic individual can apply, then define her goals. Given a generalized scoring rule of order K, we model the strategic behavior, called *vote operations*, as a set of vectors, each of which has K elements, representing the changes made to the total generalized scoring vector if the strategic individual applies this operation. We focus on integer vectors in this paper.

Definition 2 *Given a GSR GS(f, g) of order K, let $\Delta = [\vec{\delta}_1 \cdots \vec{\delta}_T]$ denote the* vote operations*, where for each $i \leq T$, $\vec{\delta}_i \in \mathbb{Z}^K$ represents the changes made to the generalized scoring vector by applying the i-th vote operation. For each $l \leq K$, let Δ_l denote the l-th row of Δ.*

We will show examples of these vote operations for some well-studied types of strategic behavior in Section 6. Given the set of available operations Δ, the strategic individual's behavior is characterized by a vector $\vec{v} \in \mathbb{N}^T_{\geq 0}$, where $\vec{v}$ is a row vector and for each $i \leq T$, v_i represents the number of i-th operation (corresponding to $\vec{\delta}_i$) that she applies. Let $(\vec{v})'$ denote the transpose of $\vec{v}$ and let $\|\vec{v}\|_1 = \sum_{i=1}^T v_i$ denote the total number of operations in $\vec{v}$, which is the L_1-norm of $\vec{v}$. It follows that $\Delta \cdot (\vec{v})'$ is the change in the total generalized scoring vector introduced by the strategic individual, where for any $l \leq K$, $\Delta_l \cdot (\vec{v})'$ is the change in the l-th component.

Next, we give definitions of the strategic individual's three goals and the corresponding computational problems studied in this paper.

Definition 3 *In the* CONSTRUCTIVE VOTE OPERATION (CVO) *problem, we are given a generalized scoring rule GS(f, g), a preference-profile P, a favored alternative c, and a set of vote operations $\Delta = [\vec{\delta}_1 \cdots \vec{\delta}_T]$, and we are asked to compute the smallest number k, denoted by* $\text{CVO}(P, c)$*, such that there exists a vector $\vec{v} \in \mathbb{N}^T_{\geq 0}$ with $\|\vec{v}\|_1 = k$ and $g\left(\mathit{Ord}(f(P) + \Delta \cdot (\vec{v})')\right) = c$. If such $\vec{v}$ does not exist, then we denote* $\text{CVO}(P, c) = \infty$.

The DESTRUCTIVE VOTE OPERATION (DVO) *problem is defined similarly, where c is the disfavored alternative, and we are asked to compute the smallest number k, denoted by* $\text{DVO}(P, c)$*, such that there exists a vector $\vec{v} \in \mathbb{N}^T_{\geq 0}$ with $\|\vec{v}\|_1 = k$ and $g\left(\mathit{Ord}(f(P) + \Delta \cdot (\vec{v})')\right) \neq c$.*

In the CHANGE-WINNER VOTE OPERATION (CWVO) *problem, we are not given c and we are asked to compute* $\text{DVO}(P, GS(f, g)(P))$*, denoted by* $\text{CWVO}(P)$.

In CVO, the strategic individual seeks to make c win; in DVO, the strategic individual seeks to make c lose; and in CWVO, the strategic individual seeks to change the current winner.

For a given instance (P, r), CWVO is a special case of DVO, where $c = \text{GS}(f, g)(P)$. We distinguish these two problems because in this paper, the input preference-profiles are generated randomly, so the winners of these preference-profiles might be different. Therefore, when the preference-profiles are randomly generated, the distribution for the solution to DVO does not immediately give us a distribution for the solution to CWVO.

4 The ILP Formulation

Let us first put aside the strategic individual's goal for the moment (i.e., making a favored alternative win, making a disfavored alternative lose, or changing the winner) and focus on the following question: given a preference-profile P and a preorder $\trianglerighteq$ over the K components of the generalized scoring vector, that is, $\trianglerighteq \in \text{Pre}(\mathcal{O}_K)$, how many vote manipulations are needed to change the order of the total generalized scoring vector to $\trianglerighteq$? Formally, given a $\text{GS}(f, g)$, a preference-profile P and $\trianglerighteq \in \text{Pre}(\mathcal{O}_K)$, we are interested in $\min\{\|\vec{v}\|_1 : \vec{v} \in \mathbb{N}^K_{\geq 0}, \text{Ord}(f(P) + \Delta \cdot (\vec{v})') = \trianglerighteq\}$.

This can be computed by the following integer linear programming $\text{ILP}_{\trianglerighteq}$, where v_i represents the ith component in $\vec{v}$, which must be a nonnegative integer. We recall that Δ_l denotes the l-th row vector of Δ.

$$
\begin{aligned}
& & \min \quad & \|\vec{v}\|_1 \\
\text{s.t.} & & \forall o_i =_{\trianglerighteq} o_j: \quad & (\Delta_i - \Delta_j) \cdot (\vec{v})' = [f(P)]_j - [f(P)]_i \\
& & \forall o_i \triangleright o_j: \quad & (\Delta_i - \Delta_j) \cdot (\vec{v})' \geq [f(P)]_j - [f(P)]_i + 1 \\
& & \forall i: \quad & v_i \geq 0
\end{aligned}
\qquad (\text{LP}_{\trianglerighteq})
$$

Now, we take the strategic individual's goal into account. We immediately have the following lemma as a warmup, whose proofs are straightforward and are thus omitted.

Lemma 1 *Given a GSR $GS(f, g)$, an alternative c, and a preference-profile P,*

• $\text{CVO}(P, c) < \infty$ *if and only if there exists* $\unrhd$ *such that* $g(\unrhd) = c$ *and* $LP_{\unrhd}$ *has an integer solution;*

• $\text{DVO}(P, c) < \infty$ *if and only if there exists* $\unrhd$ *such that* $g(\unrhd) \neq c$ *and* $LP_{\unrhd}$ *has an integer solution;*

• $\text{CWVO}(P) < \infty$ *if and only if there exists* $\unrhd$ *such that* $g(\unrhd) \neq GS(f, g)(P)$ *and* $LP_{\unrhd}$ *has an integer solution. (We do not need the input c for this problem.)*

Moreover, the solution to each of the three problems is the minimum objective value in all LPs corresponding to the problem. For example, if $\text{CVO}(P, c) < \infty$, then

$$\text{CVO}(P, c) = \min_{\|\vec{v}\|_1}\{\vec{v} \text{ is the solution to some LP}_{\unrhd} \text{ where } g(\unrhd) = c\}$$

5 The Main Theorem

In this section we prove the main theorem, which states that for a fixed m, for any generalized scoring rules and any set of vote operations Δ, if n votes are generated i.i.d., then for CVO (respectively, DVO, CWVO), with probability that can be infinitely close to 1, the solution is either 0, $\Theta(\sqrt{n})$, $\Theta(n)$, or ∞.

Theorem 1 *Let $GS(f, g)$ be an integer generalized scoring rule, let π be a distribution over all linear orders, and let Δ be a set of vote operations. Suppose we fix the number of alternatives, generate n votes i.i.d. according to π, and let P_n denote the preference-profile. Then, for any alternative c,* $\text{VO} \in \{\text{CVO}, \text{DVO}, \text{CWVO}^3\}$, *and any $\epsilon > 0$, there exists $\beta^* > 1$ such that as n goes to infinity, the total probability for the following four events sum up to more than $1 - \epsilon$: (1)* $\text{VO}(P_n, c) = 0$, *(2)* $\frac{1}{\beta^*}\sqrt{n} < \text{VO}(P_n, c) < \beta^*\sqrt{n}$, *(3)* $\frac{1}{\beta^*}n < \text{VO}(P_n, c) < \beta^* n$, *and (4)* $\text{VO}(P_n, c) = \infty$.

Proof of Theorem 1: Let $f(P_\pi) = \sum_{V \in L(\mathcal{C})} \pi(V) \cdot f(V)$, and $\unrhd_\pi = \text{Ord}(f(P_\pi))$. We first prove the theorem for CVO, and then show how to adjust the proof for DVO and CWVO. The theorem is proved in the following two steps. Step 1: we show that as n goes to infinity, with probability that goes to one we have the following: in a randomly generated P_n, the difference between any pair of components in $f(P_n)$ is either $\Theta(\sqrt{n})$ or $\Theta(n)$. Step 2: we apply sensitivity analyses to ILPs that are similar to the ILP given in Section 4 to prove that for any such preference-profile and any $\text{VO} \in \{\text{CVO}, \text{DVO}, \text{CWVO}\}$, $\text{VO}(P_n, c)$ is either 0, $\Theta(\sqrt{n})$, $\Theta(n)$, or ∞. The idea behind Step 2 is, for any preference-profile P_n, if the difference between a pair of components in $f(P_n)$ is $\Theta(\sqrt{n})$, then we consider this pair of components (not alternatives) to be "almost tied"; if the difference is $\Theta(n)$, then we consider them to be "far away". Take CVO as an example, we can easily identify the cases where $\text{CVO}(P_n, c)$ is either 0 (when $\text{GS}(f, g) = c$) or ∞ (by Lemma 1). Then, we will first try to break these "almost tied" pairs by using LPs that are similar to $\text{LP}_{\unrhd}$ introduced in Section 4, and show that if there exists an integer solution $\vec{v}$, then the objective value $\|\vec{v}\|_1$ is $\Theta(\sqrt{n})$. Otherwise, we have to change the orders between some "far away" pairs by using $\text{LP}_{\unrhd}$'s, and show that if there exists an integer solution to some $\text{LP}_{\unrhd}$ with $g(\unrhd) = c$, then the objective value is $\Theta(n)$.

Formally, given $n \in \mathbb{N}$ and $\beta > 1$, let $\mathcal{P}_\beta$ denote the set of all n-vote preference-profiles P that satisfy the following two conditions (we recall that $f(P_\pi) = \sum_{V \in L(\mathcal{C})} \pi(V) \cdot f(V)$): for any pair $i, j \leq K$,

1. if $[f(P_\pi)]_i = [f(P_\pi)]_j$ then $\frac{1}{\beta}\sqrt{n} < |[f(P)]_i - [f(P)]_j| < \beta\sqrt{n}$;
2. if $[f(P_\pi)]_i \neq [f(P_\pi)]_j$ then $\frac{1}{\beta}n < |[f(P)]_i - [f(P)]_j| < \beta n$.

The following lemma was proved in [31], which follows from the Central Limit Theorem.

Lemma 2 *For any $\epsilon > 0$, there exists β such that* $\lim_{n\to\infty} Pr(P_n \in \mathcal{P}_\beta) > 1 - \epsilon$.

[3] When $\text{VO} = \text{CWVO}$, we let $\text{VO}(P_n, c)$ denote $\text{CWVO}(P_n)$.

For any given ϵ, in the rest of the proof we fix β to be a constant guaranteed by Lemma 2. The next lemma (whose proof can be found on the author's homepage) will be frequently used in the rest of the proof.

Lemma 3 *Fix an integer matrix* $\mathbf{A}$. *There exists a constant* $\beta_{\mathbf{A}}$ *that only depends on* $\mathbf{A}$, *such that if the following LP has an integer solution, then the solution is no more than* $\beta_{\mathbf{A}} \cdot \|\vec{b}\|_\infty$.

$$\min \|\vec{x}\|_1, \text{ s.t. } \mathbf{A} \cdot \vec{x} \geq \vec{b}$$

To prove that with high probability CVO(P_n, c) is either 0, $\Theta(\sqrt{n})$, $\Theta(n)$, or ∞, we introduce the following notation. A preorder $\trianglerighteq'$ is a *refinement* of another preorder $\trianglerighteq$, if $\triangleright'$ extends $\triangleright$. That is, $\triangleright \subseteq \triangleright'$. We note that $\trianglerighteq$ is a refinement of itself. Let $\trianglerighteq' \ominus \trianglerighteq$ denote the strict orders that are in $\triangleright'$ but not in $\triangleright$. That is, $(o_i, o_j) \in (\trianglerighteq' \ominus \trianglerighteq)$ if and only if $o_i \triangleright' o_j$ and $o_i =_{\trianglerighteq} o_j$. We define the following LP that is similar to LP$_{\trianglerighteq}$ defined in Section 4, which will be used to check whether there is a way to break "almost tied" pairs of components to make c win. For any preorder $\trianglerighteq$ and any of its refinement $\trianglerighteq'$, we define LP$_{\trianglerighteq' \ominus \trianglerighteq}$ as follows.

$$\begin{array}{lrl} & \min & \|\vec{v}\|_1 \\ \text{s.t.} & \forall o_i =_{\trianglerighteq'} o_j: & (\Delta_i - \Delta_j) \cdot (\vec{v})' = [f(P)]_j - [f(P)]_i \\ & \forall (o_i, o_j) \in (\trianglerighteq' \ominus \trianglerighteq): & (\Delta_i - \Delta_j) \cdot (\vec{v})' \geq [f(P)]_j - [f(P)]_i + 1 \\ & \forall i: & v_i \geq 0 \end{array} \qquad (\text{LP}_{\trianglerighteq' \ominus \trianglerighteq})$$

LP$_{\trianglerighteq' \ominus \trianglerighteq}$ is defined with a little abuse of notation because some of its constraints depend on $\trianglerighteq$ (not only the pairwise comparisons in $(\trianglerighteq' \ominus \trianglerighteq)$). This will not cause confusion because we will always indicate $\trianglerighteq$ in the subscript. We note that there is a constraint in LP$_{\trianglerighteq' \ominus \trianglerighteq}$ for each pair of components o_i, o_j with $o_i =_{\trianglerighteq} o_j$. Therefore, LP$_{\trianglerighteq' \ominus \trianglerighteq}$ is used to find a solution that breaks ties in $\trianglerighteq$. It follows that LP$_{\trianglerighteq' \ominus \trianglerighteq}$ has an integer solution $\vec{v}$ if and only if the strategic individual can make the order between any pairs of o_i, o_j with $o_i =_{\trianglerighteq} o_j$ to be the one in $\trianglerighteq'$ by applying the i-th operation v_i times, and the total number of vote operations is $\|\vec{v}\|_1$.

The following two claims identify the preference-profiles in $\mathcal{P}_\beta$ for which CVO is $\Theta(\sqrt{n})$ and $\Theta(n)$, respectively, whose proofs can be found on the author's homepage.

Claim 1 *There exists* $N \in \mathbb{N}$ *and* $\beta' > 1$ *such that for any* $n \geq N$, *any* $P \in \mathcal{P}_\beta$, *if (1) c is not the winner for P, and (2) there exists a refinement* $\trianglerighteq^*$ *of* $\trianglerighteq_\pi = Ord(f(P_\pi))$ *such that* $g(\trianglerighteq^*) = c$ *and* $LP_{\trianglerighteq^* \ominus \trianglerighteq_\pi}$ *has an integer solution, then* $\frac{1}{\beta'}\sqrt{n} < \text{CVO}(P, c) < \beta'\sqrt{n}$.

Claim 2 *There exists* $\beta' > 1$ *such that for any* $P \in \mathcal{P}_\beta$, *if (1) c is not the winner for P, (2) there does not exist a refinement* $\trianglerighteq^*$ *of* $\trianglerighteq_\pi = Ord(f(P_\pi))$ *such that* $LP_{\trianglerighteq^* \ominus \trianglerighteq_\pi}$ *has an integer solution, and (3) there exists* $\trianglerighteq$ *such that* $g(\trianglerighteq) = c$ *and* $LP_{\trianglerighteq}$ *has an integer solution, then* $\frac{1}{\beta'} n < \text{CVO}(P, c) < \beta' n$.

Lastly, for any $P \in \mathcal{P}_\beta$ such that GS$(f, g)(P) \neq c$, the only case not covered by Claim 1 and Claim 2 is that there does not exist $\trianglerighteq$ with GS$(f, g)(\trianglerighteq) = c$ such that LP$_{\trianglerighteq}$ has an integer solution. It follows from Lemma 1 that in this case CVO$(P, c) = \infty$. We note that β' in Claim 1 and Claim 2 does not depend on n. Let β^* be an arbitrary number that is larger than the two β's. This proves the theorem for CVO.

For DVO, we only need to change $g(\trianglerighteq^*) = c$ to $g(\trianglerighteq^*) \neq c$ in Claim 1, and change $g(\trianglerighteq) = c$ to $g(\trianglerighteq) \neq c$ in Claim 2. For CWVO, CWVO(P) is never 0 and we only need to change $g(\trianglerighteq^*) = c$ to $g(\trianglerighteq^*) \neq$ GS$(f, g)(P)$ in Claim 1, and change $g(\trianglerighteq) = c$ to $g(\trianglerighteq) \neq$ GS$(f, g)(P)$ in Claim 2. ■

Remark. The intuition in Lemma 2 is quite straightforward and naturally corresponds to a random walk in multidimensional space. However, we did not find an obvious connection between random walk theory and observation made in Theorem 1 for general voting rules. We believe that it is unlikely that an obvious connection exists. One evidence is that the observation made in Theorem 1 does not hold for some voting rules. For example, consider the γ-majority rule defined in Example 2. It is not hard to see that as n goes to infinity, with probability that goes to 1 we have $\mathrm{CVO}(P_n, b) = \mathrm{DVO}(P_n, a) = \mathrm{CWVO}(P_n) = n^\gamma/2$, which is not any of the four cases described in Theorem 1 if $\gamma \neq 1/2$. (This means that for any $\gamma \neq 1/2$, γ-majority is not a generalized scoring rule, which we already know because they do not satisfy homogeneity.)

The main difficulty in proving Theorem 1 is, for generalized scoring rules we have to handle the cases where some components of the total generalized scoring vector are equivalent. This only happens with negligible probability for the randomly generated preference-profile P_n, but it is not clear how often the strategic individual can make some components equivalent in order to achieve her goal. This is the main reason for us to convert the vote manipulation problem to multiple ILPs and conduct sensitivity analyses.

6 Applications of The Main Theorem

In this section we show how to apply Theorem 1 to some well-studied types of strategic behavior, including constructive and destructive unweighted coalitional optimization, bribery and control, and margin of victory and minimum manipulation coalition size. In the sequel, we will use each subsection to define these problems and describe how they fit in our vote operation framework, and how Theorem 1 applies. In the end of the section we present a unified corollary for all these types of strategic behavior.

6.1 Unweighted Coalitional Optimization

Definition 4 *In a constructive (respectively, destructive)* UNWEIGHTED COALITIONAL OPTIMIZATION (UCO) *problem, we are given a voting rule r, a preference-profile P^{NM} of the non-manipulators, and a (dis)favored alternative $c \in \mathcal{C}$. We are asked to compute the smallest number of manipulators who can cast votes P^M such that $c = r(P^{NM} \cup P^M)$ (respectively, $c \neq r(P^{NM} \cup P^M)$).*

To see how UCO fits in the vote operation model, we view the group of manipulators as the strategic individual, and each vote cast by a manipulator is a vote operation. Therefore, the set of operations is exactly the set of all generalized scoring vectors $\{f(V) : V \in L(\mathcal{C})\}$. To apply Theorem 1, for constructive UCO we let $\mathrm{VO} = \mathrm{CVO}$ and for destructive UCO we let $\mathrm{VO} = \mathrm{DVO}$.

6.2 Bribery

In this paper we are interested in the optimization variant of the bribery problem [9].

Definition 5 *In a constructive (respectively, destructive)* OPT-BRIBERY *problem, we are given a preference-profile P and a (dis)favored alternative $c \in \mathcal{C}$. We are asked to compute the smallest number k such that the strategic individual can change no more than k votes such that c is the winner (respectively, c is not the winner).*

To see how OPT-BRIBERY falls under the vote operation framework, we view each action of "changing a vote" as a vote operation. Since the strategic individual can only change existing votes in the preference-profile, we define the set of operations to be the difference between the generalized scoring vectors of all votes and the generalized scoring vectors of votes in the support of π, that is, $\{f(V) - f(W) : V, W \in L(\mathcal{C}) \text{ s.t. } \pi(W) > 0\}$. Then, similarly the constructive variant corresponds to CVO and the destructive variant corresponds to DVO. In both cases Theorem 1 cannot be directly applied, because in the ILPs we did not limit the total number of each type of vote operations that can be used by the strategic individual. Nevertheless, we can still prove a similar proposition by taking a closer look at the relationship between CVO (DVO) and OPT-BRIBERY as follows: For any preference-profile, the solution to CVO (respectively, DVO) is a lower bound on the solution to constructive (respectively, destructive) OPT-BRIBERY, because in CVO and DVO there are no constraints on the number of each type of vote operations. We have the following four cases.

1. If the solution to CVO (DVO) is 0, then the solution to constructive (destructive) OPT-BRIBERY is also 0.

2. If the solution to CVO (DVO) is $\Theta(\sqrt{n})$, as n become large enough, with probability that goes to 1 each type of votes in the support of π will appear $\Theta(n)$, which is $> \Theta(\sqrt{n})$, times in the randomly generated preference-profile, which means that there are enough votes of each type for the strategic individual to change.

3. If the solution to CVO (DVO) is $\Theta(n)$, then the solution to constructive (destructive) OPT-BRIBERY is either $\Theta(n)$ (when the strategic individual can change *all* votes to achieve her goal), or ∞.

4. If the solution to CVO (DVO) is ∞, then the solution to constructive (destructive) OPT-BRIBERY is also ∞.

It follows that the observation made in Theorem 1 holds for OPT-BRIBERY.

6.3 Margin of Victory (MoV)

Definition 6 *Given a voting rule r and a preference-profile P, the* margin of victory (MoV) *of P is the smallest number k such that the winner can be changed by changing k votes in P. In the* MOV *problem, we are given r and P, and are asked to compute the margin of victory.*

For a given instance (P, r), MOV is equivalent to destructive OPT-BRIBERY, where $c = r(P)$. However, when the input preference-profiles are generated randomly, the winners in these profiles might be different. Therefore, the corollary of Theorem 1 for OPT-BRIBERY does not directly imply a similar corollary for MOV. This relationship is similar to the relationship between DVO and CWVO.

Despite this difference, the formulation of MOV in the vote operation framework is very similar to that of OPT-BRIBERY: The set of all operations and the argument to apply Theorem 1 are the same. The only difference is that for MOV, we obtain the corollary from the CWVO part of Theorem 1, while the corollary for OPT-BRIBERY is obtained from the CVO and DVO parts of Theorem 1.

6.4 Minimum Manipulation Coalition Size (MMCS)

The MINIMUM MANIPULATION COALITION SIZE (MMCS) problem is similar to MOV, except that in MMCS the winner must be improved for all voters who change their votes [19].

Definition 7 *In an* MMCS *problem, we are given a voting rule r and a preference-profile P. We are asked to compute the smallest number k such that a coalition of k voters can change their votes to change the winner, and all of them prefer the new winner to $r(P)$.*

Unlike MOV, MMCS falls under the vote operation framework in the following dynamic way. For each preference-profile, suppose c is the current winner. For each adversarial $d \neq c$, we use $\{f(V) - f(W) : V, W \in L(\mathcal{C}) \text{ s.t. } d \succ_W c \text{ and } \pi(W) > 0\}$ as the set of operations. That is, we only allow voters who prefer d to c to participate in the manipulative coalition. We also replace each of $\text{LP}_{\trianglerighteq}$ and $\text{LP}_{\trianglerighteq^* \ominus \trianglerighteq_\pi}$ by multiple LPs, each of which is indexed by a pair of alternatives (d, c) and the constraints are generated by using the corresponding set of operations. Then, the corollary for MMCS follows after a similar argument to that of CVO in Theorem 1.

6.5 Control by Adding/Deleting Votes (CAV/CDV)

Definition 8 *In a constructive (respectively, destructive)* OPTIMAL CONTROL BY ADDING VOTES (OPT-CAV) *problem, we are given a preference-profile P, a (dis)favored alternative $c \in \mathcal{C}$, and a set N' of additional votes. We are asked to compute the smallest number k such that the strategic individual can add k votes in N' such that c is the winner (respectively, c is not the winner).*

For simplicity, we assume that $|N'| = n$ and the votes in N' are drawn i.i.d. from a distribution π'. To show how OPT-CAV falls under the vote operation model, we let the set of operations to be the generalized scoring vectors of all votes that are in the support of π', that is, $\{f(V) : V \in L(\mathcal{C})$ and $\pi'(V) > 0\}$. Then, the corollary follows from the CVO and DVO parts of Theorem 1 via a similar argument to the argument for OPT-BRIBERY.

Definition 9 *In a constructive (respectively, destructive)* OPTIMAL CONTROL BY DELETING VOTES (OPT-CDV) *problem, we are given a preference-profile P and a (dis)favored alternative $c \in \mathcal{C}$. We are asked to compute the smallest number k such that the strategic individual can delete k votes in P such that c is the winner (respectively, c is not the winner).*

To show how OPT-CDV falls under the vote operation framework, we let the set of operations to be the negation of generalized scoring vectors of votes in the support of π', that is, $\{-f(V) : V \in L(\mathcal{C})$ and $\pi'(V) > 0\}$. Then, the corollary follows from the CVO and DVO parts of Theorem 1 via a similar argument to the argument for OPT-BRIBERY.

6.6 A Unified Corollary

The next corollary of Theorem 1 summarizes the results obtained for all types of strategic behavior studied in this section.

Corollary 1 *For any integer generalized scoring rule, any distribution π over votes, and any $X \in \big(\{$constructive, destructive$\}\times\{$UCO, OPT-BRIBERY, OPT-CAV, OPT-CDV$\}\big)\cup\{$MoV, MMCS$\}$, suppose the input preference-profiles are generated i.i.d. from π.[4] Then, for any alternative c and any $\epsilon > 0$, there exists $\beta^* > 1$ such that the total probability for the solution to X to be one of the following four cases is more than $1 - \epsilon$ as n goes to infinity: (1) 0, (2) between $\frac{1}{\beta^*}\sqrt{n}$ and $\beta^*\sqrt{n}$, (3) between $\frac{1}{\beta^*}n$ and $\beta^* n$, and (4) ∞.*

7 Acknowledgments

The author thanks Ariel Procaccia, William Zwicker, all anonymous FOCS-12 and COMSOC-12 reviewers for their very helpful comments and suggestions. This work is supported by NSF under Grant #1136996 to the Computing Research Association for the CIFellows Project.

References

[1] Eyad Baharad and Zvika Neeman. The asymptotic strategyproofness of scoring and condorcet consistent rules. *Review of Economic Design*, 4:331–340, 2002.

[2] John Bartholdi, III, Craig Tovey, and Michael Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[3] John Bartholdi, III, Craig Tovey, and Michael Trick. How hard is it to control an election? *Math. Comput. Modelling*, 16(8-9):27–40, 1992. Formal theories of politics, II.

[4] Felix Brandt. Some remarks on Dodgson's voting rule. *Mathematical Logic Quarterly*, 55:460–463, 2009.

[5] David Cary. Estimating the margin of victory for instant-runoff voting. In *Proc. EVT/WOTE*, 2011.

[6] Vincent Conitzer and Makoto Yokoo. Using mechanism design to prevent false-name manipulations. *AI Magazine*, 31(4):65–77, 2010.

[7] William J. Cook, Albertus M. H. Gerards, Alexander Schrijver, and Eva Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34(3):251–264, 1986.

[4] For CAV, the distribution over the new votes can be generated i.i.d. from a different distribution π'.

[8] Shahar Dobzinski and Ariel D. Procaccia. Frequent manipulability of elections: The case of two voters. In *Proc. WINE*, pages 653–664, 2008.

[9] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[10] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Using complexity to protect elections. *Commun. ACM*, 53:74–82, 2010.

[11] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[12] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[13] Ehud Friedgut, Gil Kalai, and Noam Nisan. Elections can be manipulated often. In *Proc. FOCS*, pages 243–249, 2008.

[14] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41:587–601, 1973.

[15] Marcus Isaksson, Guy Kindler, and Elchanan Mossel. The Geometry of Manipulation: A Quantitative Proof of the Gibbard-Satterthwaite Theorem. In *Proc. FOCS*, pages 319–328, 2010.

[16] Thomas R. Magrino, Ronald L. Rivest, Emily Shen, and David Wagner. Computing the Margin of Victory in IRV Elections. In *Proc. EVT/WOTE*, 2011.

[17] Elchanan Mossel and Miklos Z. Racz. A quantitative Gibbard-Satterthwaite theorem without neutrality. In *Proc. STOC*, 2012.

[18] Bezalel Peleg. A note on manipulability of large voting schemes. *Theory and Decision*, 11:401–412, 1979.

[19] Geoffrey Pritchard and Mark C. Wilson. Asymptotics of the minimum manipulating coalition size for positional voting rules under impartial culture behaviour. *Mathematical Social Sciences*, 1:35–57, 2009.

[20] Ariel D. Procaccia and Jeffrey S. Rosenschein. Average-case tractability of manipulation in voting via the fraction of manipulators. In *Proc. AAMAS*, pages 718–720, 2007.

[21] Jörg Rothe and Lena Schend. Typical-Case Challenges to Complexity Shields That Are Supposed to Protect Elections Against Manipulation and Control: A Survey. In *Proc. ISAIM*, 2012.

[22] Mark Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[23] Arkadii Slinko. On asymptotic strategy-proofness of classical social choice rules. *Theory and Decision*, 52:389–398, 2002.

[24] Arkadii Slinko. How large should a coalition be to manipulate an election? *Mathematical Social Sciences*, 47(3):289–293, 2004.

[25] Philip B. Stark. Conservative statistical post-election audits. *The Annals of Applied Statistics*, 2(2):550–581, 2008.

[26] Philip B. Stark. A sharper discrepancy measure for post-election audits. *The Annals of Applied Statistics*, 2(3):982–985, 2008.

[27] Philip B. Stark. Efficient post-election audits of multiple contests: 2009 California tests. In *4th Annual Conference on Empirical Legal Studies (CELS)*, 2009.

[28] Philip B. Stark. Risk-limiting post-election audits: P-values from common probability inequalities. *IEEE Transactions on Information Forensics and Security*, 4:1005–1014, 2009.

[29] Philip B. Stark. Super-simple simultaneous single-ballot risk-limiting audits. In *Proceedings of 2010 EVT/WOTE Conference*, 2010.

[30] T. Nicolaus Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[31] Lirong Xia. Computing the margin of victory for various voting rules. In *Proc. EC*, 2012.

[32] Lirong Xia and Vincent Conitzer. Generalized scoring rules and the frequency of coalitional manipulability. In *Proc. EC*, pages 109–118, 2008.

[33] Lirong Xia and Vincent Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *Proc. EC*, pages 99–108, 2008.

[34] Lirong Xia and Vincent Conitzer. Finite local consistency characterizes generalized scoring rules. In *Proc. IJCAI*, pages 336–341, 2009.

[35] Lirong Xia, Vincent Conitzer, and Ariel D. Procaccia. A scheduling approach to coalitional manipulation. In *Proc. EC*, pages 275–284, 2010.

[36] Michael Zuckerman, Omer Lev, and Jeffrey S. Rosenschein. An algorithm for the coalitional manipulation problem under maximin. In *Proc. AAMAS*, 2011.

[37] Michael Zuckerman, Ariel D. Procaccia, and Jeffrey S. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392–412, 2009.

Lirong Xia
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA
Email: lxia@seas.harvard.edu